ENE TECHNOLOGY INC.

KB926D

Keyboard Controller
Data Sheet

Revision 2.0
May 2009

Headquarters
4F-1, No.9, Prosperity Rd.,
Science-based Industrial Park,
Hsinchu City, Taiwan, R.O.C
TEL: 886-3-6662888
FAX: 886-3-6662999
http://www.ene.com.tw

Taipei Office
4F, No.88, Bauchiau Rd.
Shindian City, Taipei,
Taiwan, R.O.C.
TEL: 886-2-89111525
FAX: 886-2-89111523

ENE Technology Inc.
Confidential Document
Restricted Circulation

- This document is issued for _____ only. Please do not transfer it to other companies.
- This document is the property of ENE Technology Corp. It should be returned once unused.
- Please do not make any copy of this document and deliver to others.

THIS DOCUMENT CONTAINS CONFIDEMTIAL INFORMATION OF ENE PRODUCTS. ANY UNAUTHORIZED USE OR DISCLOSURE COULD IMPACT ENE'S COMPETITIVE ADVANTAGE.

This document issued to:	
Recipient	
Company	
Serial Number	

Revision

Revision	Description	Date
0.1	1. First Release	2008/4
0.2	1. Fix some ESB description, decoding range…etc 2. ECCFG[1] reserved	2008/4
0.3	1. Change part number to D1	2008/4
0.4	1. Update power-on state of GPIO0B(Pin 17)	2008/5
0.5	1. The register PLLCFG reset value is 0x70 and the output frequency is 38MHz	2008/6
0.6	1. Adding programming model/sample code	2008/7
0.7	1. Adding interrupt enable/pending flag table In 4.18.1.3 2. Adding PCICLK and CLKRUN# description in 4.1.3 3. Part number update	2008/8
0.8	1. Adding more description for CLKRUN# and PCICLK	2008/8
0.9	1. Update I/O cell characteristics	2008/9
1.0	1. Update I/O cell information 2. Update “EnE” to “ENE” 3. Restore ECCFG[1]	2008/10
1.1	1. Add a suggestion on I/O Cell OCT04H application 2. Update the reset default states of SPICS#(Pin 128), SPICLK(Pin 126), MOSI(Pin 120), MISO(Pin 119) and PCIRST#(Pin 13) 3. Add new part number KB926QFD3 4. Remove section 4.12.4 and 4.12.5 5. Add 5.7.1 SPI Flash Timing spec.	2008/12
1.2	1. Support BGA package 2. Correct the signal names from GPI42/GPI43/GPXAxx/GPXDx to AD4/AD5/GPXIOAxx/GPXIODx on section 2.1	2009/1
1.3	1.Correct typo GPIXO into GPXIO on Ball-map&Pin-Assignment	2009/2
1.4	1.Fix description of REG LPCSCFG, 2.ESB decode Range 4	2009/2
1.5	1.Fix description about GPWU	2009/2
1.6	1.Fix clock domain ratio	2009/3
1.7	1.Fix GPIO structure description	2009/3
1.8	1.Fix SMBSTS[5] Description	2009/4
1.9	1.Add application notice to I/O cell about A/D D/A pads 2.Fix PS2 ports naming rule the same as software	2009/5
2.0	1.Modify for detail SPI timing chart	2009/5

CONTENT

1. General Description

1.1 Overview

The KB926D is an embedded controller with LPC interface to connect with host. The embedded controller contains industrial standard 8051 microprocessor and provides function of i8042 keyboard controller basically. The KB926D is designed with Shared-ROM architecture with SPI flash. The EC firmware and system BIOS will exist in one SPI flash. The embedded controller also features rich interfaces for applications, such as PS/2 interface, Keyboard Matrix, PWM, A/D converter, D/A converter, Fan controller, SMBus controller, GPIO controllers and extension interface for future applications. The chapter 1.2 highlights of all features in the KB926D.

1.2 Features

LPC Low Pin Count Interface

- SIRQ supporting IRQ1, IRQ12, SCI or SMI# interrupt and one programmable IRQ provided.
- I/O Address Decoding:
  - Legacy KBC I/O port 60h/64h
  - Programmable EC I/O port, 62h/66h(recommend)
  - I/O port 68h/6Ch (sideband)
  - **2** Programmable 4-byte Index-I/O ports to access internal EC registers.
  - **1** Programmable extended (debug) port I/O.
- Memory Decoding:
  - Firmware Hub decode
  - LPC memory decode
- Compatible with LPC specification v1.1

X-bus Bus Interface (XBI) : Flash Interface

- SPI flash is supported, size up to 4MB.
- SPI frequency supports 33/45/66MHz.
- New SPI command (dual read) to enhance the performance.
- The 64KB code memory can be mapped into system memory by one 16KB and one 48KB programmable pages independently.
- Support SPI flash in-system-programming via IKB pins.
- Enhanced pre-fetch mechanism.

8051 Microprocessor

- Compatible with industrial 8051 instructions with 3 cycles.
- 8051 runs at 8/16/22 MHz, programmable.
- 128 bytes internal RAM.
- 24 extended interrupt sources.
- Two 16-bit timers.
- Full duplex UART integrated.
- Supports idle and stop mode.
- Enhanced embedded debug interface.

8042 Keyboard Controller

- 8 standard 8042 commands processed by hardware.
- Each hardware command can be optionally processed by firmware.
- Pointing device multiplex mode support.
- Fast GA20 and KB reset support.

PS/2 Controller

- Support at most 3 external PS/2 devices.
- External PS/2 device operation in firmware mode.

Internal Keyboard Matrix (IKB)

- 18x8 keyboard scan matrix.
- Support W2K Internet and multimedia keys.
- Support hotkey events defined.
- Ghost key cancellation mechanism provided.

Embedded Controller (EC)

- ACPI Spec 2.0 compliant.
- 5 standard EC command supported directly by hardware.
- Each hardware command can be processed by firmware optionally.
- Programmable EC I/O ports, 62h/66h by default.

SMBus Host Controller

- SMBus Spec 2.0 compliant.
- SMBus with 2 interfaces.
- Byte mode support.
- Slave function support.

Digital-to-Analog Converter (DAC)

- 4 DAC channels with 8-bit resolution.
- All DAC pins can be alternatively configured as GPO (general purpose output) function.

Analog-to-Digital Converter (ADC)

- 6 ADC channels with 10-bit resolution.
- All ADC pins can be alternatively configured as GPI (general purpose input) function.

Pulse Width Modulator (PWM)

- 6 PWM channels are provided. (8-bit *2, 14-bit *2 and FANPWM(12-bit) *2)
- Clock source selectable:.
  - 1MHz/64KHz/4KHz/256Hz (for 8-bit PWM)
  - Peripheral clock or 1MHz (for 14-bit PWM)
  - Peripheral clock (for FANPWM)
- Duty cycle programmable and cycle time up to 1 sec(for 8-bit PWM)

WatchDog Timer (WDT)

- 32.768KHz input clock.
- 8-bit counter with 128ms unit for watchdog reset.
- Two watchdog reset mechanism.
  - Reset 8051
  - Reset whole chip, except GPIO.

Real Time Clock

- 32.768KHz input clock.
- 24-bit timer support.

General Purpose Timer (GPT)

- Two 16-bit and two 8-bit general purpose timer with 32.768KHz clock source.

General Purpose Wakeup (GPWU)

- Those I/O with GPI (general purpose input) configuration can generate interrupts or wakeup events, except those pins named in **GPXIOAxx**.

General Purpose Input/Output (GPIO)

- All general purpose I/O can be programmed as input or output.
- All output pins can be configured to be tri-state optionally.
- All input pins are equipped with pull-up, high/low active and edge/level trigger selection.
- For the pins of DAC can be configured as GPO function only.
- For the pins of ADC can be configured as GPI function only.
- A specific pair of GPIO pins with signal pass-through feature.

FAN Controller

- Two fan controllers with tachometer inputs.
- Automatic fan control support.
- 12-bit FANPWM support.

Consumer IR (CIR)

- Several protocols decoded/encoded by hardware.
- Interrupt for CIR application.
- Support wide/narrow band receiver.
- Transmit/Receive simultaneously.
- Remote power-on support.

ENE Serial Bus Interface (ESB)

- A proprietary and flexible interface for extension with ENE KBC.
- Firmware accesses ESB devices via internal memory address directly.
- Interrupt capability.

ENE Debug Interface (EDI)

- Flexible debug interface with SPI pins.
- Keil-C development tool compatible

SPI Device Interface (SDI)

- A simple SPI master controller is embedded in the KBC.
- Flexible design for SPI applications.

Power Management

- Sleep mode: 8051 program counter (PC) stops and enters idle mode.
- Deep sleep mode: All clocks stop except external 32.768KHz OSC. 8051 enters stop mode.

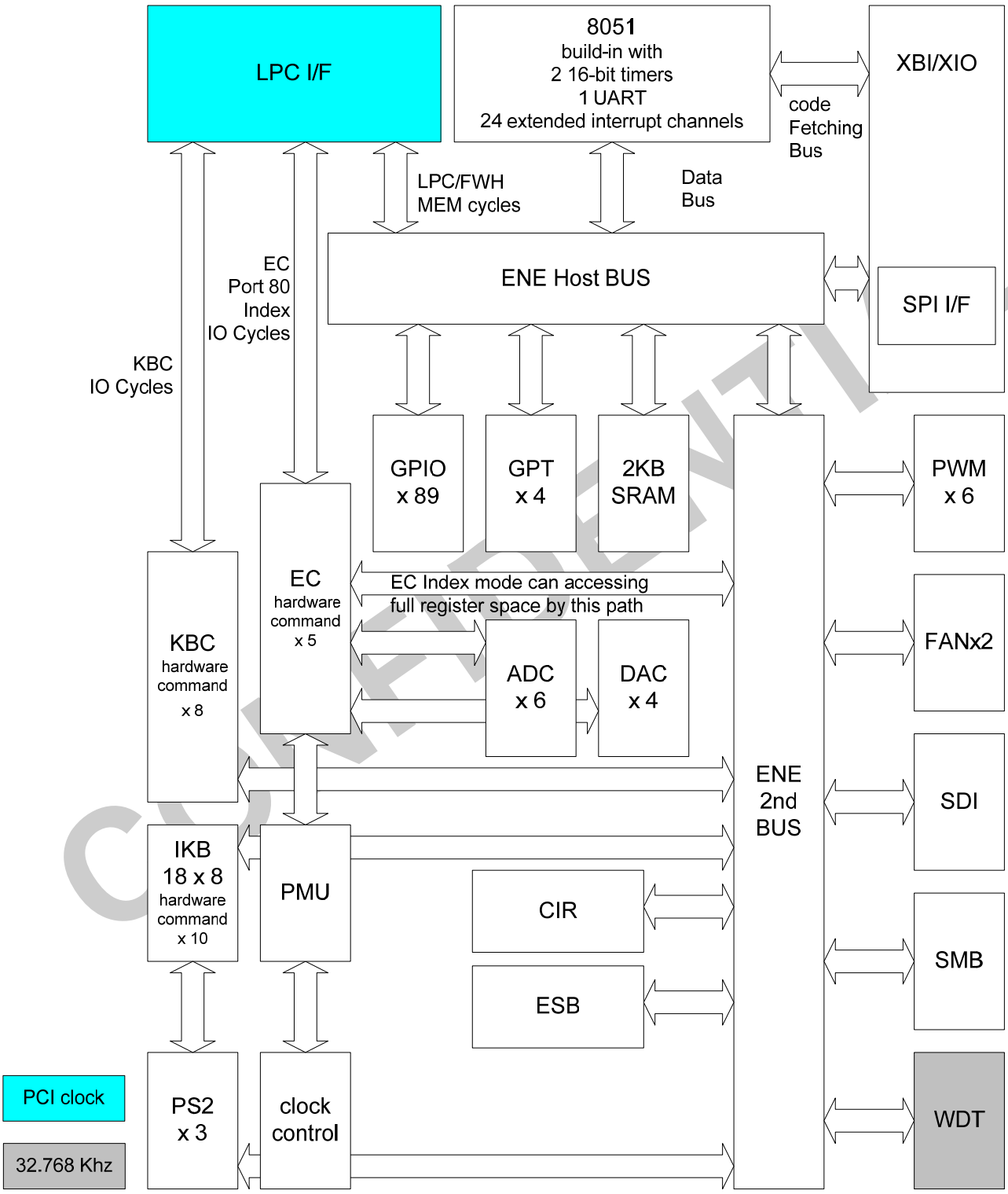
Package

- 128-pin LQFP package, Lead Free (RoHS).
- 128-ball LFBGA package, Lead Free (RoHS).

1.3 Comparison (KB926C vs. KB926D)

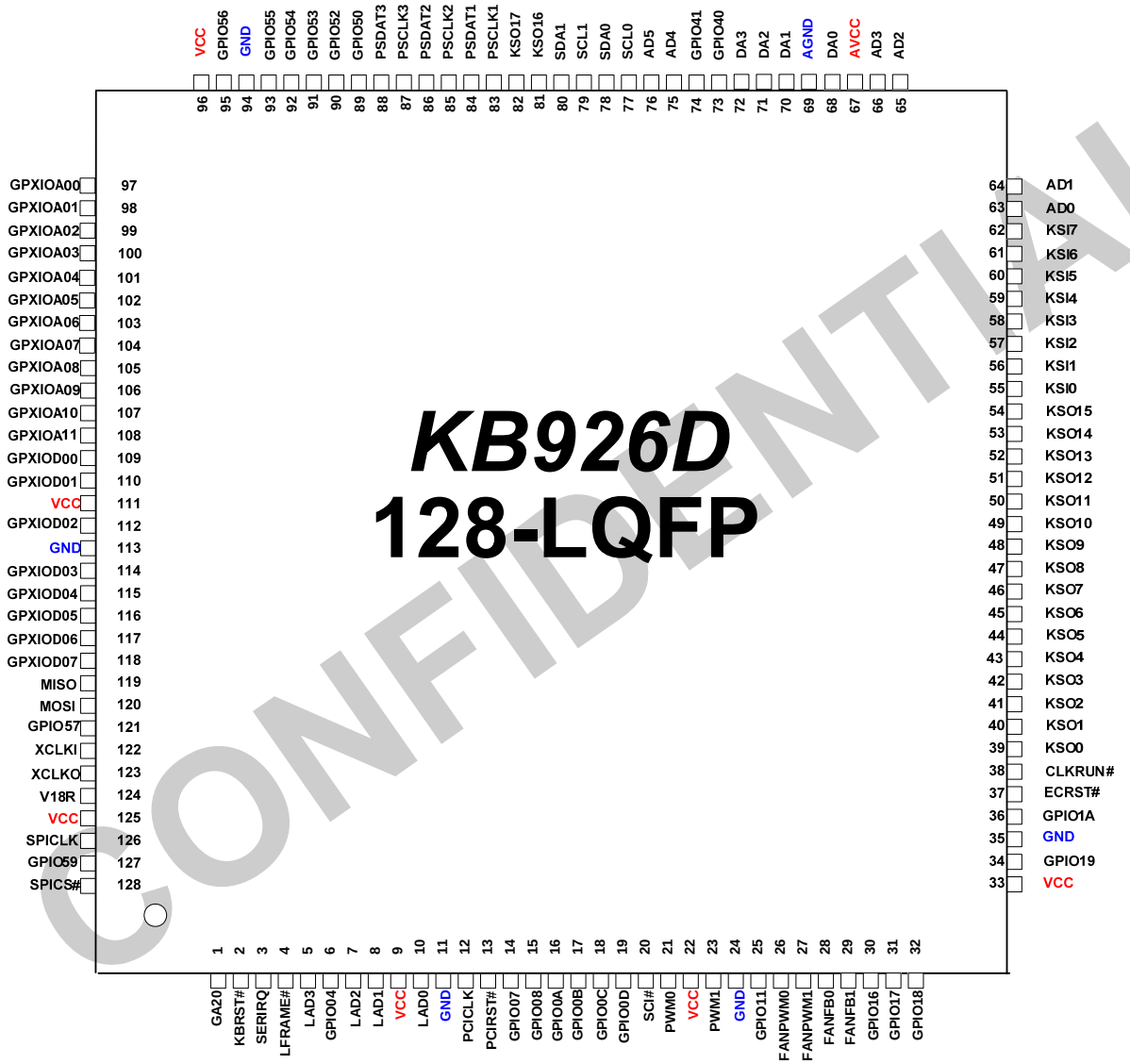
	KB926C	KB926D
Microprocessor	8051	8051
Built-in SRAM	2KB	2KB
LPC	2 index-I/O sets	2 index-I/O sets
X-Bus	SPI ROM: 4MB	SPI ROM: 4MB **Enhanced pre-fetch mechanism.**
Real Time Clock	support	Support
ADC	Six 10-bit ADC channels	Six 10-bit ADC channels
DAC	Four 8-bit DAC channels	Four 8-bit DAC channels
WDT	20 bit	20 bit
PWM	6 sets PWM0/1 – 8 bit PWM2/3 – 14 bit FANPWM0/1 – 12 bit	6 sets PWM0/1 – 8 bit PWM2/3 – 14 bit FANPWM0/1 – 12 bit
External PS/2 I/F	3	3
GPIO	Programmable Bi-direction I/O GPIO pass through : 1 pair Max GPIO: 100(926C)	Programmable Bi-direction I/O GPIO pass through : 1 pair Max GPIO: 100(926D)
IKB Matrix	18x8	18x8
FAN controller	2	2
GPT	4	4
SMBus	2 Byte mode support	2 Byte mode support
CIR	Hardware encode/decode IRQ and I/O port support Carrier frequency calculation TX with carrier modulation Learning mode support	Hardware encode/decode IRQ and I/O port support Carrier frequency calculation TX with carrier modulation Learning mode support **TX/RX simultaneously**
EDI	None	**Support**
ESB	Support	**Support and Enhanced**
SDI	Support	Support
Package	128 LQFP	128 LQFP
Dimension	14mmx14mm	14mmx14mm

1.4 Block Diagram

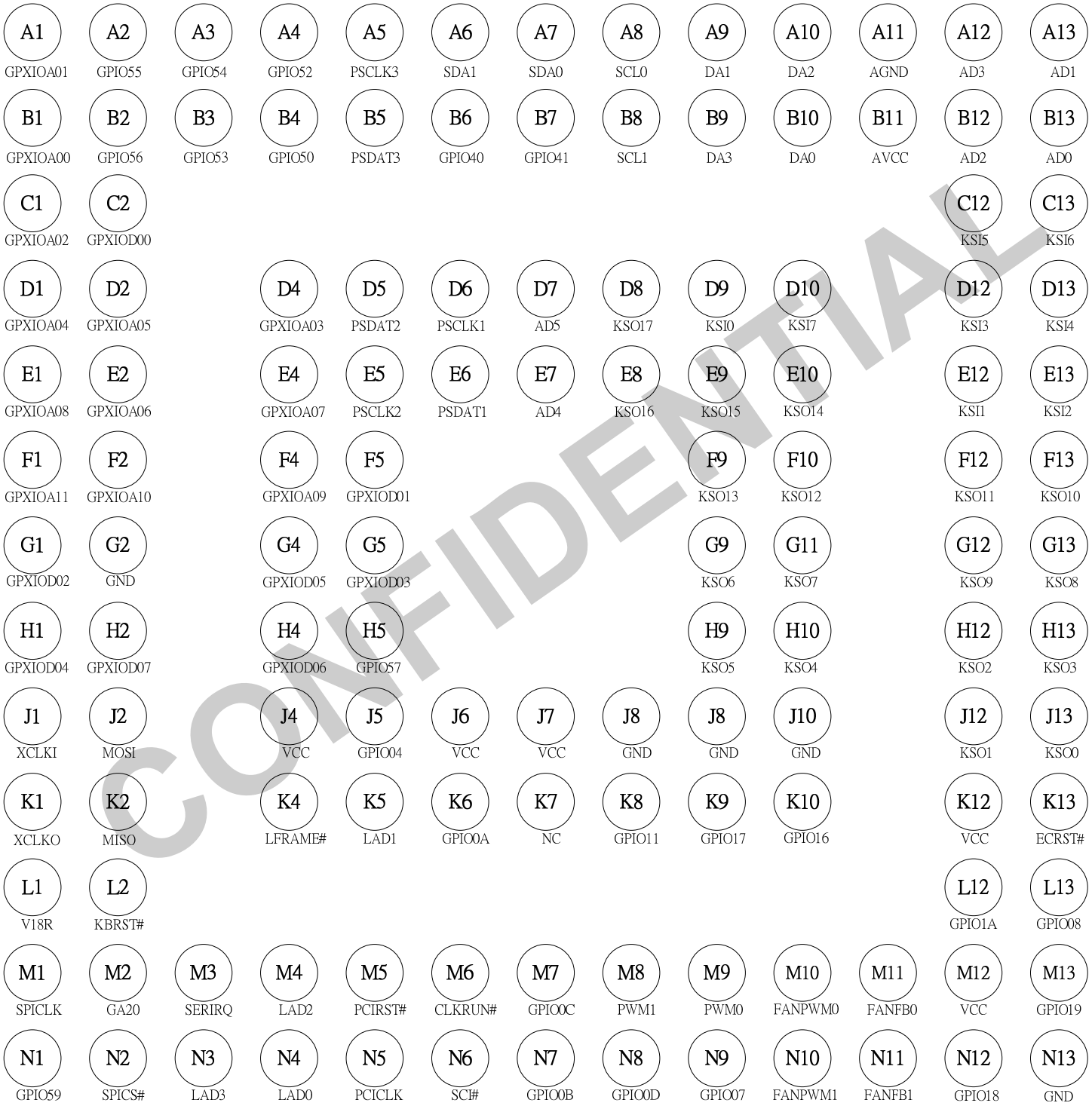


2. Pin Assignment and Description

2.1 KB926D 128-pin LQFP Diagram Top View



2.2 KB926D 128 LFBGA Ball Map



2.3 KB926D Pin Assignment Side A

KB926D Pin No.	KB926D BGA	Name	GPIO	Alt Output	Alt. Input	Default	ECRST# L/H	IO CELL
1	M2	GA20		GA20		GPIO00	HiZ / HiZ	BQC04HIV
2	L2	KBRST#		KBRST#		GPIO01	HiZ / HiZ	BQC04HIV
3	M3	SERIRQ					HiZ / HiZ	BCC16HI
4	K4	LFRAME#					HiZ / HiZ	BCC16HI
5	N3	LAD3					HiZ / HiZ	BCC16HI
6	J5	GPIO04				GPIO04	HiZ / HiZ	BQC04HIV
7	M4	LAD2					HiZ / HiZ	BCC16HI
8	K5	LAD1					HiZ / HiZ	BCC16HI
9	J7/K12/M12	VCC						VCC
10	N4	LAD0					HiZ / HiZ	BCC16HI
11	J8/J9/N13	GND						GND
12	N5	PCICLK					IE/IE	BCC16HI
13	M5	PCIRST#			PCIRST#	GPIO05	HiZ / IE	BCC16HI
14	N9	GPIO07	GPIO07	i_clk_8051		GPIO07	HiZ / HiZ	BQC04HIV
15	L13	GPIO08	GPIO08	i_clk_peri		GPIO08	HiZ / HiZ	BQC04HIV
16	K6	GPIO0A	GPIO0A		RLC_RX2	GPIO0A	HiZ / HiZ	BQC04HIV
17	N7	GPIO0B	GPIO0B	ESB_CLK		GPIO0B	PU / PU	BQCW16HIV
18	M7	GPIO0C	GPIO0C	ESB_DAT_O	ESB_DAT_I	GPIO0C	HiZ / HiZ	BQCW16HIV
19	N8	GPIO0D	GPIO0D	RLC_TX2		GPIO0D	HiZ / HiZ	BQC04HIV
20	N6	SCI#	GPIO0E	SCI#		GPIO0E	HiZ / HiZ	BQC04HIV
21	M9	PWM0	GPIO0F	PWM0		GPIO0F	HiZ / HiZ	BQC04HI
22	K7	VCC	VCC					VCC
23	M8	PWM1	GPIO10	PWM1		GPIO10	HiZ / HiZ	BQC04HI
24	J10	GND	GND					GND
25	K8	GPIO11	GPIO11	PWM2		GPIO11	HiZ / HiZ	BQC04HIV
26	M10	FANPWM0	GPIO12	FANPWM0		GPIO12	HiZ / HiZ	BQC04HI
27	N10	FANPWM1	GPIO13	FANPWM1		GPIO13	HiZ / HiZ	BQC04HI
28	M11	FANFB0	GPIO14		FANFB0	GPIO14	HiZ / HiZ	BQC04HI
29	N11	FANFB1	GPIO15		FANFB1	GPIO15	HiZ / HiZ	BQC04HI
30	K10	GPIO16	GPIO16	E51TXD		GPIO16	HiZ / HiZ	BQC04HI
31	K9	GPIO17	GPIO17	E51CLK	E51RXD	GPIO17	HiZ / HiZ	BQC04HI
32	N12	GPIO18	GPIO18			GPIO18	HiZ / HiZ	BQC04HIV

2.4 KB926D Pin Assignment Side B

KB926D Pin No.	KB926D BGA	Name	GPIO	Alt Output	Alt. Input	Default	ECRST# L/H	IO CELL
33	J7/K12/M12	VCC						VCC
34	M13	GPIO19	GPIO19	PWM3		GPIO19	HiZ / HiZ	BCC16HI
35	J8/J9/N13	GND						GND
36	L12	GPIO1A	GPIO1A	NUMLED#		GPIO1A	HiZ / HiZ	BCC16HI
37	K13	ECRST#					IE / IE	BQC04HIV
38	M6	CLKRUN#	GPIO1D	CLKRUN#	CLKRUN#	GPIO1D	HiZ / HiZ	BCC16HI
39	J13	KSO0	GPIO20	KSO0	TP_TEST	GPIO20	IE(PU)/IE(PU)	BQC04HIV
40	J12	KSO1	GPIO21	KSO1	TP_PLL	GPIO21	IE(PU)/IE(PU)	BQC04HIV
41	H12	KSO2	GPIO22	KSO2	TP_ANA_TEST	GPIO22	IE(PU)/IE(PU)	BQC04HIV
42	H13	KSO3	GPIO23	KSO3	TP_ISP	GPIO23	IE(PU)/IE(PU)	BQC04HIV
43	H10	KSO4	GPIO24	KSO4		GPIO24	HiZ / HiZ	BQC04HIV
44	H9	KSO5	GPIO25	KSO5		GPIO25	HiZ / HiZ	BQC04HIV
45	G9	KSO6	GPIO26	KSO6		GPIO26	HiZ / HiZ	BQC04HIV
46	G10	KSO7	GPIO27	KSO7		GPIO27	HiZ / HiZ	BQC04HIV
47	G13	KSO8	GPIO28	KSO8		GPIO28	HiZ / HiZ	BQC04HIV
48	G12	KSO9	GPIO29	KSO9		GPIO29	HiZ / HiZ	BQC04HIV
49	F13	KSO10	GPIO2A	KSO10		GPIO2A	HiZ / HiZ	BQC04HIV
50	F12	KSO11	GPIO2B	KSO11		GPIO2B	HiZ / HiZ	BQC04HIV
51	F10	KSO12	GPIO2C	KSO12		GPIO2C	HiZ / HiZ	BQC04HIV
52	F9	KSO13	GPIO2D	KSO13		GPIO2D	HiZ / HiZ	BQC04HIV
53	E10	KSO14	GPIO2E	KSO14		GPIO2E	HiZ / HiZ	BQC04HIV
54	E9	KSO15	GPIO2F	KSO15	E51_RXD(ISP)	GPIO2F	HiZ / HiZ	BQC04HIV
55	D9	KSI0	GPIO30	E51_TXD(ISP)	KSI0	GPIO30	IE(PU)/IE(PU)	BQC04HIV
56	E12	KSI1	GPIO31		KSI1	GPIO31	IE(PU)/IE(PU)	BQC04HIV
57	E13	KSI2	GPIO32		KSI2	GPIO32	IE(PU)/IE(PU)	BQC04HIV
58	D12	KSI3	GPIO33		KSI3	GPIO33	IE(PU)/IE(PU)	BQC04HIV
59	D13	KSI4	GPIO34		KSI4/EDI_CS	GPIO34	IE(PU)/IE(PU)	BQC04HIV
60	C12	KSI5	GPIO35		KSI5/EDI_CLK	GPIO35	IE(PU)/IE(PU)	BQC04HIV
61	C13	KSI6	GPIO36		KSI6/EDI_DIN	GPIO36	IE(PU)/IE(PU)	BQC04HIV
62	D10	KSI7	GPIO37	EDI_DO	KSI7	GPIO37	IE(PU)/IE(PU)	BQC04HIV
63	B13	AD0	**GPI**38		AD0	GPI38	HiZ / HiZ	IQTHI
64	A13	AD1	**GPI**39		AD1	GPI39	HiZ / HiZ	IQTHI

2.5 KB926D Pin Assignment Side C

KB926D Pin No.	KB926D BGA	Name	GPIO	Alt Output	Alt. Input	Default	ECRST# L/H	IO CELL
65	B12	AD2	**GPI**3A		AD2	**GPI**3A	HiZ / HiZ	IQTHI
66	A12	AD3	**GPI**3B		AD3	**GPI**3B	HiZ / HiZ	IQTHI
67	B11	AVCC						AVCC
68	B10	DA0	**GPO**3C	DA0		**GPO**3C	HiZ / HiZ	OCT04H
69	A11	AGND						AGND
70	A9	DA1	**GPO**3D	DA1		**GPO**3D	HiZ / HiZ	OCT04H
71	A10	DA2	**GPO**3E	DA2		**GPO**3E	HiZ / HiZ	OCT04H
72	B9	DA3	**GPO**3F	DA3		**GPO**3F	HiZ / HiZ	OCT04H
73	B6	GPIO40	GPIO40		CIR_RX	GPIO40	HiZ / HiZ	BQC04HI
74	B7	GPIO41	GPIO41	CIR_RLC_TX		GPIO41	HiZ / HiZ	BQC04HIV
75	E7	AD4	**GPI**42		AD4	**GPI**42	HiZ / HiZ	IQTHI
76	D7	AD5	**GPI**43		AD5	**GPI**43	HiZ / HiZ	IQTHI
77	A8	SCL0	GPIO44	SCL0		GPIO44	HiZ / HiZ	BQC04HI
78	A7	SDA0	GPIO45	SDA0		GPIO45	HiZ / HiZ	BQC04HI
79	B8	SCL1	GPIO46	SCL1		GPIO46	HiZ / HiZ	BQC04HI
80	A6	SDA1	GPIO47	SDA1		GPIO47	HiZ / HiZ	BQC04HI
81	E8	KSO16	GPIO48	KSO16		GPIO48	HiZ / HiZ	BQC04HIV
82	D8	KSO17	GPIO49	KSO17		GPIO49	HiZ / HiZ	BQC04HIV
83	D6	PSCLK1	GPIO4A	PSCLK1/P80CLK		GPIO4A	HiZ / HiZ	BQC04HI
84	E6	PSDAT1	GPIO4B	PSDAT1P80DAT		GPIO4B	HiZ / HiZ	BQC04HI
85	E5	PSCLK2	GPIO4C	PSCLK1		GPIO4C	HiZ / HiZ	BCC16HI
86	D5	PSDAT2	GPIO4D	PSDAT1		GPIO4D	HiZ / HiZ	BCC16HI
87	A5	PSCLK3	GPIO4E	PSCLK2		GPIO4E	HiZ / HiZ	BQC04HI
88	B5	PSDAT3	GPIO4F	PSDAT2		GPIO4F	HiZ / HiZ	BQC04HI
89	B4	GPIO50	GPIO50			GPIO50	HiZ / HiZ	BQC04HI
90	A4	GPIO52	GPIO52	E51CS#		GPIO52	HiZ / HiZ	BCC16HI
91	B3	GPIO53	GPIO53	CAPSLED#	E51TMR1	GPIO53	HiZ / HiZ	BCC16HI
92	A3	GPIO54	GPIO54	WDT_LED#	E51TMR0	GPIO54	HiZ / HiZ	BCC16HI
93	A2	GPIO55	GPIO55	SCROLED#	E51INT0	GPIO55	HiZ / HiZ	BCC16HI
94	J8/J9/N13	GND						GND
95	B2	GPIO56	GPIO56		E51INT1	GPIO56	HiZ / HiZ	BQC04HIV
96	J7/K12/M12	VCC						VCC

2.6 KB926D Pin Assignment Side D

KB926D Pin No.	KB926D BGA	Name	GPIO	Alt Output	Alt. Input	Default	ECRST# L/H	IO CELL
97	B1	GPXIOA00	GPXIOA00	SDICS#			HiZ / HiZ	BQC04HIV
98	A1	GPXIOA01	GPXIOA01	SDICLK			HiZ / HiZ	BQC04HIV
99	C1	GPXIOA02	GPXIOA02	SDIMOSI			HiZ / HiZ	BQC04HIV
100	D4	GPXIOA03	GPXIOA03				HiZ / HiZ	BQC04HIV
101	D1	GPXIOA04	GPXIOA04				HiZ / HiZ	BQC04HIV
102	D2	GPXIOA05	GPXIOA05				HiZ / HiZ	BQC04HIV
103	E2	GPXIOA06	GPXIOA06				HiZ / HiZ	BQC04HIV
104	E4	GPXIOA07	GPXIOA07				HiZ / HiZ	BQC04HIV
105	E1	GPXIOA08	GPXIOA08				HiZ / HiZ	BQCZ16HIV
106	F4	GPXIOA09	GPXIOA09				HiZ / HiZ	BQCZ16HIV
107	F2	GPXIOA10	GPXIOA10				HiZ / HiZ	BQCZ16HIV
108	F1	GPXIOA11	GPXIOA11				HiZ / HiZ	BQCZ16HIV
109	C2	GPXIOD00	GPXIOD00		SDIMISO		HiZ / HiZ	BQC04HIV
110	F5	GPXIOD01	GPXIOD01				HiZ / HiZ	BQC04HIV
111	J6	VCC					HiZ / HiZ	VCC
112	G1	GPXIOD02	GPXIOD02				HiZ / HiZ	BQC04HIV
113	G2	GND					HiZ / HiZ	GND
114	G5	GPXIOD03	GPXIOD03				HiZ / HiZ	BQC04HIV
115	H1	GPXIOD04	GPXIOD04				HiZ / HiZ	BQC04HIV
116	G4	GPXIOD05	GPXIOD05				HiZ / HiZ	BQC04HIV
117	H4	GPXIOD06	GPXIOD06				HiZ / HiZ	BQC04HIV
118	H2	GPXIOD07	GPXIOD07				HiZ / HiZ	BQC04HIV
119	K2	MISO			MISO	MISO	HiZ / IE	BQC04HI
120	J2	MOSI		MOSI		MOSI	HiZ / Ox	BQCZ16HIV
121	H5	GPIO57	GPIO57	XCLK32K		GPIO57	HiZ / HiZ	BQC04HIV
122	J1	XCLKI						
123	K1	XCLKO						
124	L1	V18R						
125	J4	VCC						VCC
126	M1	SPICLK	GPIO58	SPICLK		SPICLK	HiZ / Ox	BQCW16HIV
127	N1	GPIO59	GPIO59		TEST_CLK SPICLKI	GPIO59	IE / IE	BQC04HIV
128	N2	SPICS#		SPICS#		SPICS#	HiZ / Ox	BQCZ16HIV

2.7 I/O Cell Descriptions

2.7.1 I/O Buffer Table

Cell	Description	Application
BQCZ16HIV	Schmitt trigger, 16mA Output / Sink Current, Input / Output / Pull Up Enable(40K Ω), 5 V Tolerance.	GPIO
BQC04HIV	Schmitt trigger, 4mA Output / Sink Current, Input / Output / Pull Up Enable(40K Ω), 5 V Tolerance	GPIO
BQCW16HIV	Schmitt trigger, 16mA Output / Sink Current, 5 V Tolerance, Input / Output / Pull Up Enable	ESB_CLK/ ESB_DAT/ SPI_CLK
BCC16HI	16mA Output / Sink Current , 5 V Tolerance, Input / Output Enable	LPC I/F
BQC04HI	Schmitt trigger, 4mA Output / Sink Current, 5 V Tolerance, Input / Output Enable	GPIO
IQTHI	Mixed mode IO, ADC Enable, with GPI, Input Enable	ADC, GPI
OCT04H	Mixed mode IO, DAC Enable, with GPO, 4mA Output Current, Output Enable **(For GPO function, it is not recommended to control the device powered before KBC chip.)**	DAC, GPO

*** 5V Tolerance, only if pull-high disable and output disable.**
**** Please note, the total current in each side of chip can not exceed over 48mA.**

Application Notice: The Pads with I/O cells of IQTHI, OCT04H should be designed carefully. Under specific environment which: KBC is power-off, external application circuit is power-on.
Signals mustn't be connected with pads of IQTHI/OCT04H (ADCs/DACs). It would cause unexpected voltage level on these pad if KBC is still power-off.

3. Pin Descriptions

3.1 Hardware Trap

Hardware trap pins are used to latch external signal at rising edge of **ECRST#**. The hardware trap pins are for some special purpose which should be defined while boot-up. The following table gives the collection of hardware trap pins. Please note, all the following hardware trap pins are **pull-high** internally after reset.

Trap Name	Pin No.	Description
TP_TEST	39	While this trap is asserted to be low, the internal DPLL circuit uses other clock source for reference, instead of 32KHz oscillator. **Low:** test clock mode enable **High:** normal mode using 32KHz oscillator.
TP_PLL	40	While this trap is asserted to be low, some DPLL related signals can be output for test. **Low:** DPLL test mode enable. **High:** DPLL test mode disable
TP_ANA_TEST	41	While this trap is asserted to be low, some ADC related signals can be output for test. **Low:** ADC test mode enable. ＊ **High:** ADC test mode disable
TP_ISP	42	While this trap is asserted to be low, SPI Flash can be programmed via RS232 I/F, i.e., TX and RX. Please note, while entering ISP mode, the TX/RX pins are linked to GPIO30/GPIO2F **Low:** SPI flash programming in ISP mode enable ＊ **High**: SPI flash programming in ISP mode disable

＊ Please note while TP_ANA_TEST and TP_ISP keep low at the same time, a mechanism called **FlashDirectAccess** will enable. That is, users can flush and program a SPI flash via specific IKB pins with external tool.

FlashDirectAccess:

The KBC provides a new interface to program SPI flash via IKB interface. With this feature, users can easily utilize 4 pins from keyboard matrix (IKB) without disassembly whole machine. These 4 pins are connected directly to external SPI programmer. The following table shows the pins mapping while entering FlashDirectAccess mode.

Pin No.	Normal Mode	FlashDirectAccess Mode	Comment
47	KSO8 (O)	FDA_SPICLK (I)	
48	KSO9 (O)	FDA_SPICS (I)	
49	KSO10 (O)	FDA_SPIDI (I)	
50	KSO11 (O)	FDA_SPIDO (O)	

3.2 Pin Descriptions by Functions

3.2.1 Low Pin Count I/F Descriptions.

Pin Name	Pin No.	Direction	Description
LAD[3:0]	5, 7,8,10	I/O	LPC address bus.
LFARAME#	4	I	LPC frame control signal.
PCIRST#	13	I	LPC module reset by this signal.
PCICLK	12	I	33MHz PCI clock input.
SERIRQ	3	I/O	Serial IRQ
CLKRUN#	38	I/OD	Clock run control

3.2.2 SPI Flash I/F Descriptions

Pin Name	Pin No.	Direction	Description
MISO	119	I	SPI read control signal
MOSI	120	O	SPI write control signal
SPICLK	126	O	SPI clock output
SPICS#	128	O	SPI chip select signal
These pins are input/output disable during reset phase.			

3.2.3 PS/2 I/F Descriptions

Pin Name	Pin No.	Direction	Description
PSCLK1	83	I/OD	PS/2 port 1 clock
PSDAT1	84	I/OD	PS/2 port 1 data
PSCLK2	85	I/OD	PS/2 port 2 clock
PSDAT2	86	I/OD	PS/2 port 2 data
PSCLK3	87	I/OD	PS/2 port 3 clock
PSDAT3	88	I/OD	PS/2 port 3 data

3.2.4 Internal Keyboard Encoder (IKB) Descriptions

Pin Name	Pin No.	Direction	Description
KSO[17:0]	82,81,54-39	O	Keyboard Scan Out
KSI[7:0]	62-55	I	Keyboard Scan In

3.2.5 SMBus Descriptions

Pin Name	Pin No.	Direction	Description
SCL0	77	I/OD	SMBus clock (interface 0)
SDA0	78	I/OD	SMBus data (interface 0)
SCL1	79	I/OD	SMBus clock (interface 1)
SDA1	80	I/OD	SMBus data (interface 1)

3.2.6 FAN Descriptions

Pin Name	Pin No.	Direction	Description
FANPWM0	26	O	FANPWM0 output
FANPWM1	27	O	FANPWM1 output
FANFB0	28	I	FAN0 tachometer input
FANFB1	29	I	FAN1 tachometer input

3.2.7 Pulse Width Modulation (PWM) Descriptions

Pin Name	Pin No.	Direction	Description
PWM[1:0]	23, 21	O	PWM pulse output

3.2.8 Analog-to-Digital Converter Descriptions

Pin Name	Pin No.	Direction	Description
AD[3:0]	66-63	I	10bit A/D converter input
AD[5:4]	76,75	I	10bit A/D converter input

3.2.9 Digital-to-Analog Converter Descriptions

Pin Name	Pin No.	Direction	Description
DA[3:0]	72-70,68	O	8bit D/A converter output

3.2.10 8051 External I/F Descriptions

Pin Name	Pin No.	Direction	Description
E51TXD	30	O	8051 serial port, transmit port.
E51RXD	31	I	8051 serial port, receive port.
E51CLK	31	O	For different serial scheme, E51CLK will shift out clock.
E51CS#	90	O	
E51TMR0	92	I	
E51INT0	93	I	
E51TMR1	91	I	
E51INT1	95	I	

3.2.11 External Clock Descriptions

Pin Name	Pin No.	Direction	Description
XCLKI	122	I	32.768KHz input
XCLKO	123	O	32.768KHz output

3.2.12 Miscellaneous Signals Descriptions

Pin Name	Pin No.	Direction	Description
GA20	1	O	KBC will gate A20 address line
KBRST#	2	O	KBRST# is used to generate system reset.
SCI#	20	O	SCI# asserts to the system for requesting service while related events occur.
ECRST#	37	I	While ECRST# asserted, the KBC will reset globally.

3.2.13 Power Pins Descriptions

Pin Name	Pin No.	Direction	Description
VCC	9,22,33,96,111,125		Power supply for digital circuit.
GND	11,24,35,94,113		Power ground for digital circuit.
AVCC	67		Power supply for analog circuit.
AGND	69		Power ground for analog circuit.

4. Module Descriptions

4.1 Chip Architecture

4.1.1 Power Planes

Two power planes are in the KBC. One is for digital logic and the other is for analog circuit. Both power planes are ±10% tolerance for recommend operation condition, The KBC provides V1.8 power plane for different generation.

Power Plane	Description	Power	Ground
Digital Plane	This power provides power for all digital logic no matter what power mode is.	VCC	GND
Analog Plane	This power provides power for all analog logic, such as A/D and D/A converter.	AVCC	AGND
Digital V1.8	The system inputs 3.3V power and the internal regulator outputs 1.8V voltage. The 1.8V output should connect a **capacitor** for stable purpose.	V1.8	GND

4.1.2 Clock Domains

Three clock sources, PCICLK, DPLL_CLK and XCLKI will be discussed in this section. A summary is list in the following table.

Clock	Description
PCICLK	PCI clock 33MHz for LPC I/F.
DPLL_CLK	Main clock for 8051/peripheral. DPLL clock can be generated with or without XCLK for reference. DPLL clock can be divided for different applications. Fig. 4-1 gives an example for illustration.
XCLKI	External 32.768KHz for reference.

The following figure shows more detail about the operation in the KBC. The external 32.768KHz is provided for two purposes. One is to provide an accurate reference for internal DPLL module, and the other one is to provide another clock source for watchdog timer.

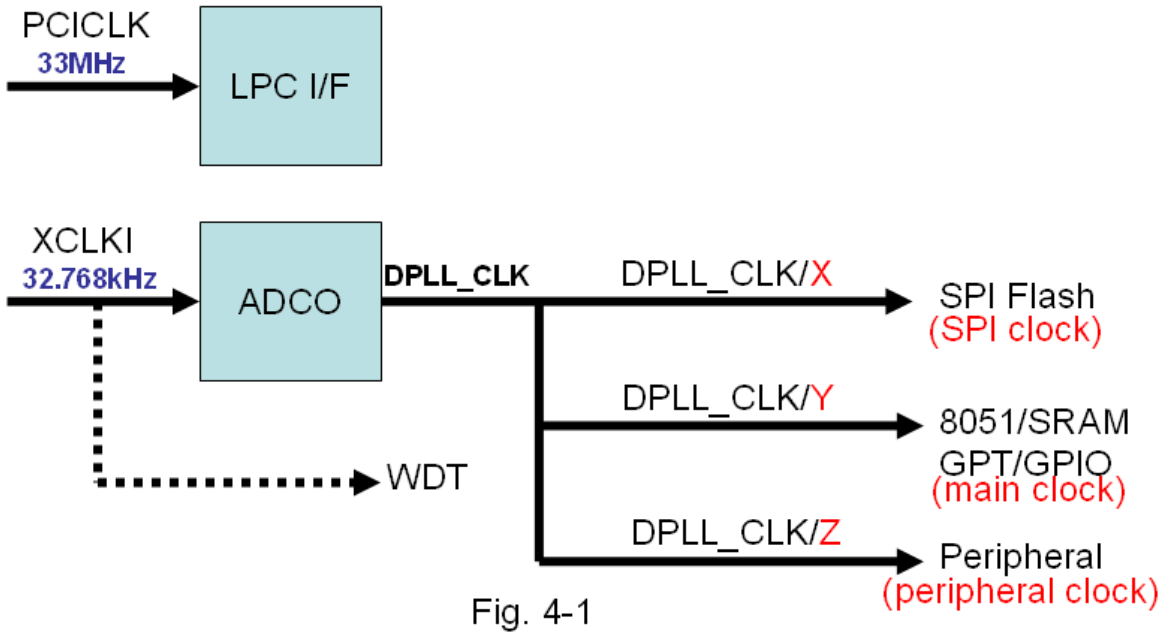


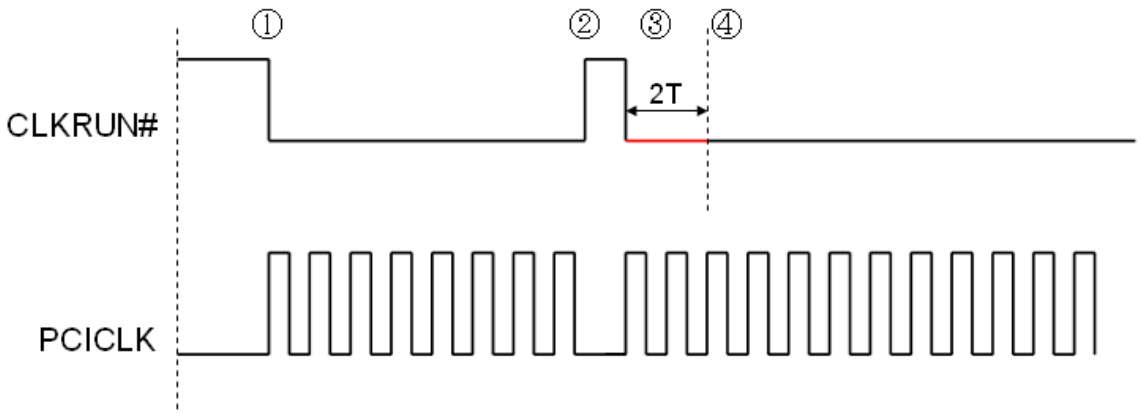
Fig. 4-1

The possible (X,Y,Z) combination is summarized as the following table.

	CLKCFG[6]=0(default)	CLKCFG[6]=1
CLKCFG[3:2]=0(default)	(X,Y,Z)=(4,8,16) *	(X,Y,Z)=(1,8,16)
CLKCFG[3:2]=1	(X,Y,Z)=(2,4,8)	(X,Y,Z)=(1,4,8)
CLKCFG[3:2]=2	(X,Y,Z)=(2,3,6)	(X,Y,Z)=(1,3,6)
CLKCFG[3:2]=3	(X,Y,Z)=(2,2,4)	(X,Y,Z)=(1,2,4)
* While power on default, no matter what value of CLKCFG[3:2] and CLKCFG[6] are, the (X,Y,Z) is always (4, 8, 16)		

4.1.3 PCICLK and CLKRUN#

While system power-on, the host starts to drive CLKRUN# low for a while to inform the slaves that a 33MHz PCICLK will be given. At this moment, CLKRUN# of KBC is in input mode. If the host tries to stop the PCICLK for some purpose, the CLKRUN# will be de-asserted. In the current design, the KBC needs PCICLK for normal operation. Therefore the KBC keeps CLKRUN# for 2 clock cycles and releases it. This forces the host to start driving PCICLK. The following figure gives the explanation. For more detail please refer to *PCI Mobile Design Guide version 1.1*.



① Host asserts CLKRUN# and PCICLK is driven
② Host de-asserts CLKRUN# for some considerations
③ KBC monitors CLKRUN# de-asserting and then KBC keeps asserting CLKRUN#. This forces PCICLK keeping driving.
④ Host monitors CLKRUN# for 3T and sees the request from device. And then Host keeps CLKRUN# asserting.

4.1.4 Internal Memory Map

No	Module	Descriptions	Address Range	Size (Byte)	
1	Flash	Space mapped to system BIOS	0x0000~0xF3FF	61K	
2	XRAM	Embedded SRAM	0xF400~0xFBFF	2K	
3	GPIO	General purpose I/O	0xFC00~0xFC7F	128	1K
4	KBC	Keyboard controller	0xFC80~0xFC8F	16	
5	ESB	ENE serial bus controller	0xFC90~0xFC9F	16	
6	IKB	Internal keyboard matrix	0xFCA0~0xFCAF	16	
7	RSV	Reserved	0xFCB0~0xFCBF	16	
8	RSV	Reserved	0xFCC0~0xFCCF	16	
9	RSV	Reserved	0xFCD0~0xFCDF	16	
10	RSV	Reserved	0xFCE0~0xFCEF	16	
11	RSV	Reserved	0xFCF0~0xFDFF	272	
12	PWM	Pulse width modulation	0xFE00~0xFE1F	32	
13	FAN	Fan controller	0xFE20~0xFE4F	48	
14	GPT	General purpose timer	0xFE50~0xFE6F	32	
15	SDI	SPI device interface	0xFE70~0xFE7F	16	
16	WDT	Watchdog timer	0xFE80~0xFE8F	16	
17	LPC	Low pin count interface	0xFE90~0xFE9F	16	
18	XBI	X-bus interface	0xFEA0~0xFEBF	32	
19	CIR	Consumer IR controller	0xFEC0~0xFECF	16	
20	RSV	Reserved	0xFED0~0xFEDFh	16	
21	PS2	PS/2 interface	0xFEE0~0xFEFF	32	
22	EC	Embedded controller	0xFF00~0xFF1F	32	
23	GPWU	General purpose wakeup event	0xFF20~0xFF7F	96	
24	SMBus	System management bus controller	0xFF80~0xFFFF	128	

4.2 GPIO

4.2.1 GPIO Function Description

The GPIO module is flexible for different applications. Each GPIO pin can be configured as alternative input or alternative output mode. The alternative function can be selected by register setting. A summary table is given as below for more detail.

GPIO	Alt. Output	Alt. Input	Default Alt. Output	Alt. Selection Reg.
GPIO00	GA20		GPIO00	GPIOFS00.[0]
GPIO01	KBRST#		GPIO01	GPIOFS00.[1]
GPIO02*			GPIO02	GPIOFS00.[2]
GPIO03*			GPIO03	GPIOFS00.[3]
GPIO04			GPIO04	GPIOFS00.[4]
GPIO05		PCIRST#	GPIO05	GPIOFS00.[5]
GPIO06*			GPIO06	GPIOFS00.[6]
GPIO07	i_clk(8051)		GPIO07	GPIOFS00.[7]
GPIO08	i_clk(peripheral)		GPIO08	GPIOFS08.[0]
GPIO09*			GPIO09	GPIOFS08.[1]
GPIO0A		RLC_RX2	GPIO0A	GPIOFS08.[2]
GPIO0B	ESB_CLK		GPIO0B	GPIOFS08.[3]
GPIO0C	ESB_DAT	ESB_DAT_I	GPIO0C	GPIOFS08.[4]
GPIO0D	RLC_TX2		GPIO0D	GPIOFS08.[5]
GPIO0E	SCI#		GPIO0E	GPIOFS08.[6]
GPIO0F	PWM0		GPIO0F	GPIOFS08.[7]
GPIO10	PWM1		GPIO10	GPIOFS10.[0]
GPIO11	PWM2		GPIO11	GPIOFS10.[1]
GPIO12	FANPWM0		GPIO12	GPIOFS10.[2]
GPIO13	FANPWM1		GPIO13	GPIOFS10.[3]
GPIO14		FANFB0	GPIO14	GPIOFS10.[4]
GPIO15		FANFB1	GPIO15	GPIOFS10.[5]
GPIO16	E51TXD		GPIO16	GPIOFS10.[6]
GPIO17	E51CLK	E51RXD	GPIO17	GPIOFS10.[7]
GPIO18			GPIO18	GPIOFS18.[0]
GPIO19	PWM3		GPIO19	GPIOFS18.[1]
GPIO1A	NUMLED#		GPIO1A	GPIOFS18.[2]
GPIO1B*			GPIO1B	GPIOFS18.[3]
GPIO1C*			GPIO1C	GPIOFS18.[4]
GPIO1D	CLKRUN#	CLKRUN#	GPIO1D	GPIOFS18.[5]
GPIO1E*			GPIO1E	GPIOFS18.[6]
GPIO1F*			GPIO1F	GPIOFS18.[7]
GPIO20	KSO00	TP_TEST	GPIO20	GPIOFS20.[0]
GPIO21	KSO01	TP_PLL	GPIO21	GPIOFS20.[1]
GPIO22	KSO02	TP_ANA_TEST	GPIO22	GPIOFS20.[2]
GPIO23	KSO03	TP_ISP	GPIO23	GPIOFS20.[3]

GPIO	Alt. Output	Alt. Input	Default Alt. Output	Alt. Selection Reg.
GPIO24	KSO04		GPIO24	GPIOFS20.[4]
GPIO25	KSO05		GPIO25	GPIOFS20.[5]
GPIO26	KSO06		GPIO26	GPIOFS20.[6]
GPIO27	KSO07		GPIO27	GPIOFS20.[7]
GPIO28	KSO08		GPIO28	GPIOFS28.[0]
GPIO29	KSO09		GPIO29	GPIOFS28.[1]
GPIO2A	KSO10		GPIO2A	GPIOFS28.[2]
GPIO2B	KSO11		GPIO2B	GPIOFS28.[3]
GPIO2C	KSO12		GPIO2C	GPIOFS28.[4]
GPIO2D	KSO13		GPIO2D	GPIOFS28.[5]
GPIO2E	KSO14		GPIO2E	GPIOFS28.[6]
GPIO2F	KSO15	E51_RXD(ISP)	GPIO2F	GPIOFS28.[7]
GPIO30	E51_TXD(ISP)	KSI0	GPIO30	GPIOFS30.[0]
GPIO31		KSI1	GPIO31	GPIOFS30.[1]
GPIO32		KSI2	GPIO32	GPIOFS30.[2]
GPIO33		KSI3	GPIO33	GPIOFS30.[3]
GPIO34		KSI4	GPIO34	GPIOFS30.[4]
GPIO35		KSI5	GPIO35	GPIOFS30.[5]
GPIO36		KSI6	GPIO36	GPIOFS30.[6]
GPIO37		KSI7	GPIO37	GPIOFS30.[7]
GPI38		AD0		GPIOFS38.[0]
GPI39		AD1		GPIOFS38.[1]
GPI3A		AD2		GPIOFS38.[2]
GPI3B		AD3		GPIOFS38.[3]
GPO3C	DA0		GPIO3C	GPIOFS38.[4] ★
GPO3D	DA1		GPIO3D	GPIOFS38.[5] ★
GPO3E	DA2		GPIO3E	GPIOFS38.[6] ★
GPO3F	DA3		GPIO3F	GPIOFS38.[7] ★
GPIO40		CIR_RX	GPIO40	GPIOFS40.[0]
GPIO41	CIR_RLC_TX		GPIO41	GPIOFS40.[1]
GPI42		AD4		GPIOFS40.[2]
GPI43		AD5		GPIOFS40.[3]
GPIO44	SCL0		GPIO44	GPIOFS40.[4]
GPIO45	SDA0		GPIO45	GPIOFS40.[5]
GPIO46	SCL1		GPIO46	GPIOFS40.[6]
GPIO47	SDA1		GPIO47	GPIOFS40.[7]
GPIO48	KSO16		GPIO48	GPIOFS48.[0]
GPIO49	KSO17		GPIO49	GPIOFS48.[1]
GPIO4A	PSCLK1/P80CLK		GPIO4A	GPIOFS48.[2]
GPIO4B	PSDAT1/P80DAT		GPIO4B	GPIOFS48.[3]
GPIO4C	PSCLK2		GPIO4C	GPIOFS48.[4]
GPIO4D	PSDAT2		GPIO4D	GPIOFS48.[5]

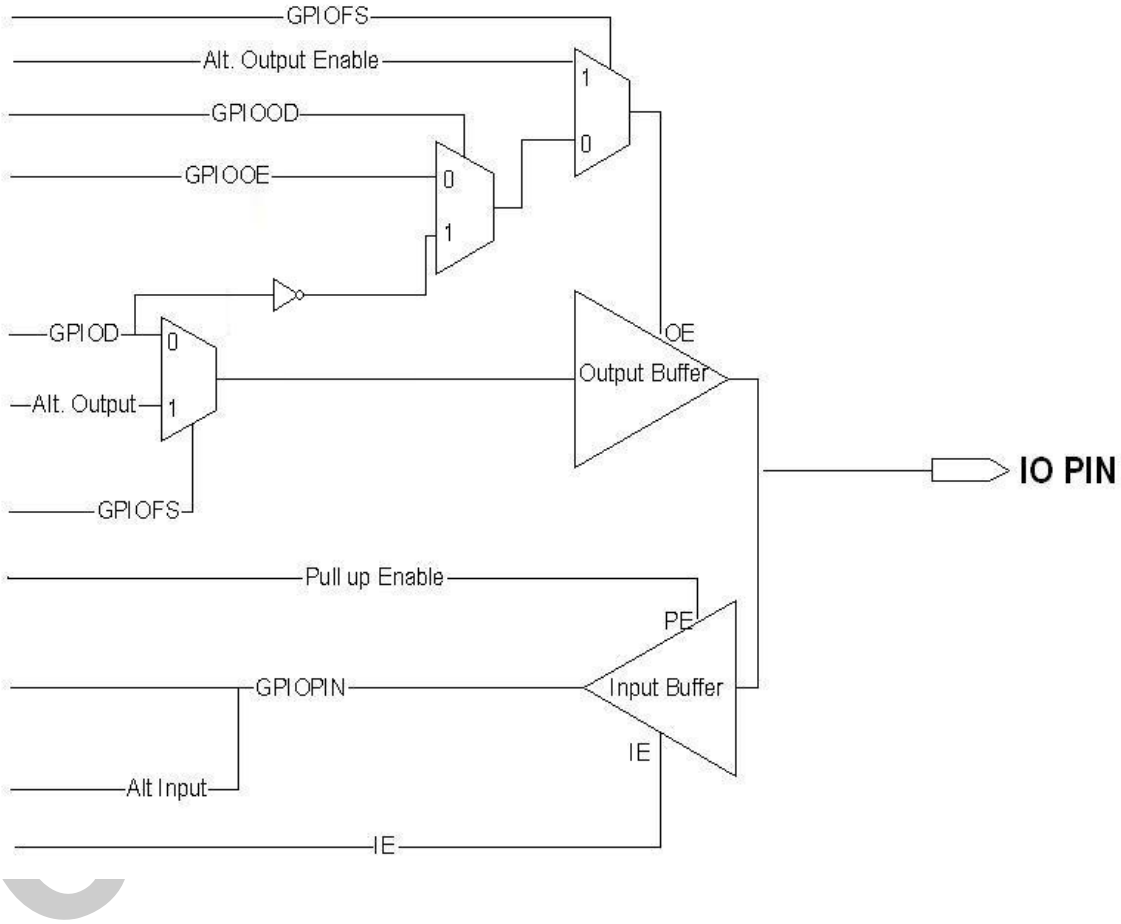
GPIO	Alt. Output	Alt. Input	Default Alt. Output	Alt. Selection Reg.
GPIO4E	PSCLK3		GPIO4E	GPIOFS48.[6]
GPIO4F	PSDAT3		GPIO4F	GPIOFS48.[7]
GPIO50			GPIO50	GPIOFS50.[0]
GPIO51*			GPIO51	GPIOFS50.[1]
GPIO52	E51CS#		GPIO52	GPIOFS50.[2]
GPIO53	CAPSLED#	E51TMR1	GPIO53	GPIOFS50.[3]
GPIO54	WDT_LED#	E51TMR0	GPIO54	GPIOFS50.[4]
GPIO55	SCORLED#	E51INT0	GPIO55	GPIOFS50.[5]
GPIO56		E51INT1	GPIO56	GPIOFS50.[6]
GPIO57	XCLK32K		GPIO57	GPIOFS50.[7]
GPIO58	SPICLK		GPIO58	GPIOFS58.[0]
GPIO59		TEST_CLK/SPICLK	GPIO59	GPIOFS58.[1]
GPXIOA00	SDICS#			GPIO_MISC.[2]
GPXIOA01	SDICLK			GPIO_MISC.[2]
GPXIOA02	SDIDO			GPIO_MISC.[2]
GPXIOA03				
GPXIOA04				
GPXIOA05				
GPXIOA06				
GPXIOA07				
GPXIOA08				
GPXIOA09				
GPXIOA10				
GPXIOA11				
GPXIOA12*				
GPXIOA13*				
GPXIOA14*				
GPXIOA15*				
GPXIOA16*				
GPXIOA17*				
GPXIOA18*				
GPXIOD00		SDIDI		
GPXIOD01				
GPXIOD02				
GPXIOD03				
GPXIOD04				
GPXIOD05				
GPXIOD06				
GPXIOD07				

* In KB926D, these GPIO pins no more exist. The corresponding register bits do not work.

★ If DAC function selected, please do not set this register bit.

4.2.2 GPIO Structures

In this section, the GPIO structure is illustrated as following diagram. The upper part is alternative output circuit and the lower part is alternative input circuit. In the figure, GPIOFS is used to enable alternative output. GPIOOD is for open-drain setting with output function. GPIOOE is the switch for data output. As shown in the figure, the alternative input embedded with pull-high and interrupt feature.



4.2.3 GPIO Attribution Table

GPIO	Alt. Output	Alt. Input	Default Alt. Output	Alt. Selection Reg.	Input Enable	Output Enable	Pull Up (40KΩ)	Open Drain	Output Current
GPIO00	GA20		GPIO00	GPIOFS00.[0]	V	V	V	V	2-4mA
GPIO01	KBRST#		GPIO01	GPIOFS00.[1]	V	V	V	V	2-4mA
GPIO02*			GPIO02	GPIOFS00.[2]					
GPIO03*			GPIO03	GPIOFS00.[3]					
GPIO04			GPIO04	GPIOFS00.[4]	V	V	V	V	2-4mA
GPIO05		PCIRST#	GPIO05	GPIOFS00.[5]	V	V		V	8-16mA
GPIO06*			GPIO06	GPIOFS00.[6]					
GPIO07	i_clk(8051)		GPIO07	GPIOFS00.[7]	V	V	V	V	2-4mA
GPIO08	i_clk(peripheral)		GPIO08	GPIOFS08.[0]	V	V	V	V	2-4mA
GPIO09*			GPIO09	GPIOFS08.[1]					
GPIO0A		RLC_RX2	GPIO0A	GPIOFS08.[2]	V	V	V	V	2-4mA
GPIO0B	ESB_CLK		GPIO0B	GPIOFS08.[3]	V	V	V	V	8-16mA
GPIO0C	ESB_DAT	ESB_DAT_I	GPIO0C	GPIOFS08.[4]	V	V	V	V	8-16mA
GPIO0D	RLC_TX2		GPIO0D	GPIOFS08.[5]	V	V	V	V	2-4mA
GPIO0E	SCI#		GPIO0E	GPIOFS08.[6]	V	V	V	V	2-4mA
GPIO0F	PWM0		GPIO0F	GPIOFS08.[7]	V	V		V	2-4mA
GPIO10	PWM1		GPIO10	GPIOFS10.[0]	V	V		V	2-4mA
GPIO11	PWM2		GPIO11	GPIOFS10.[1]	V	V	V	V	2-4mA
GPIO12	FANPWM0		GPIO12	GPIOFS10.[2]	V	V		V	2-4mA
GPIO13	FANPWM1		GPIO13	GPIOFS10.[3]	V	V		V	2-4mA
GPIO14		FANFB0	GPIO14	GPIOFS10.[4]	V	V		V	2-4mA
GPIO15		FANFB1	GPIO15	GPIOFS10.[5]	V	V		V	2-4mA
GPIO16	E51TXD		GPIO16	GPIOFS10.[6]	V	V		V	2-4mA
GPIO17	E51CLK	E51RXD	GPIO17	GPIOFS10.[7]	V	V		V	2-4mA
GPIO18			GPIO18	GPIOFS18.[0]	V	V	V	V	2-4mA
GPIO19	PWM3		GPIO19	GPIOFS18.[1]	V	V		V	8-16mA
GPIO1A	NUMLED#		GPIO1A	GPIOFS18.[2]	V	V		V	8-16mA
GPIO1B*			GPIO1B	GPIOFS18.[3]					
GPIO1C*			GPIO1C	GPIOFS18.[4]					
GPIO1D	CLKRUN#	CLKRUN#	GPIO1D	GPIOFS18.[5]	V	V	V	V	8-16mA
GPIO1E*			GPIO1E	GPIOFS18.[6]					
GPIO1F*			GPIO1F	GPIOFS18.[7]					
GPIO20	KSO00	TP_TEST	GPIO20	GPIOFS20.[0]	V	V	V	V	2-4mA
GPIO21	KSO01	TP_PLL	GPIO21	GPIOFS20.[1]	V	V	V	V	2-4mA
GPIO22	KSO02	TP_ANA_TEST	GPIO22	GPIOFS20.[2]	V	V	V	V	2-4mA
GPIO23	KSO03	TP_ISP	GPIO23	GPIOFS20.[3]	V	V	V	V	2-4mA
GPIO24	KSO04		GPIO24	GPIOFS20.[4]	V	V	V	V	2-4mA
GPIO25	KSO05		GPIO25	GPIOFS20.[5]	V	V	V	V	2-4mA
GPIO26	KSO06		GPIO26	GPIOFS20.[6]	V	V	V	V	2-4mA
GPIO27	KSO07		GPIO27	GPIOFS20.[7]	V	V	V	V	2-4mA
GPIO28	KSO08		GPIO28	GPIOFS28.[0]	V	V	V	V	2-4mA
GPIO29	KSO09		GPIO29	GPIOFS28.[1]	V	V	V	V	2-4mA
GPIO2A	KSO10		GPIO2A	GPIOFS28.[2]	V	V	V	V	2-4mA
GPIO2B	KSO11		GPIO2B	GPIOFS28.[3]	V	V	V	V	2-4mA
GPIO2C	KSO12		GPIO2C	GPIOFS28.[4]	V	V	V	V	2-4mA
GPIO2D	KSO13		GPIO2D	GPIOFS28.[5]	V	V	V	V	2-4mA
GPIO2E	KSO14		GPIO2E	GPIOFS28.[6]	V	V	V	V	2-4mA
GPIO2F	KSO15	E51_RXD(ISP)	GPIO2F	GPIOFS28.[7]	V	V	V	V	2-4mA

GPIO	Alt. Output	Alt. Input	Default Alt. Output	Alt. Selection Reg.	Input Enable	Output Enable	Pull Up (40KΩ)	Open Drain	Output Current
GPIO30	E51_TXD(ISP)	KSI0	GPIO30	GPIOFS30.[0]	V	V	V	V	2-4mA
GPIO31		KSI1	GPIO31	GPIOFS30.[1]	V	V	V	V	2-4mA
GPIO32		KSI2	GPIO32	GPIOFS30.[2]	V	V	V	V	2-4mA
GPIO33		KSI3	GPIO33	GPIOFS30.[3]	V	V	V	V	2-4mA
GPIO34		KSI4	GPIO34	GPIOFS30.[4]	V	V	V	V	2-4mA
GPIO35		KSI5	GPIO35	GPIOFS30.[5]	V	V	V	V	2-4mA
GPIO36		KSI6	GPIO36	GPIOFS30.[6]	V	V	V	V	2-4mA
GPIO37		KSI7	GPIO37	GPIOFS30.[7]	V	V	V	V	2-4mA
GPI38		AD0		GPIOFS38.[0]	V				
GPI39		AD1		GPIOFS38.[1]	V				
GPI3A		AD2		GPIOFS38.[2]	V				
GPI3B		AD3		GPIOFS38.[3]	V				
GPO3C	DA0		GPO3C	GPIOFS38.[4]		V			2-4mA
GPO3D	DA1		GPO3D	GPIOFS38.[5]		V			2-4mA
GPO3E	DA2		GPO3E	GPIOFS38.[6]		V			2-4mA
GPO3F	DA3		GPO3F	GPIOFS38.[7]		V			2-4mA
GPIO40		CIR_RX	GPIO40	GPIOFS40.[0]	V	V		V	2-4mA
GPIO41	CIR_RLC_TX		GPIO41	GPIOFS40.[1]	V	V	V	V	2-4mA
GPI42		AD4		GPIOFS40.[2]	V				2-4mA
GPI43		AD5		GPIOFS40.[3]	V				2-4mA
GPIO44	SCL0		GPIO44	GPIOFS40.[4]	V	V		V	2-4mA
GPIO45	SDA0		GPIO45	GPIOFS40.[5]	V	V		V	2-4mA
GPIO46	SCL1		GPIO46	GPIOFS40.[6]	V	V		V	2-4mA
GPIO47	SDA1		GPIO47	GPIOFS40.[7]	V	V		V	2-4mA
GPIO48	KSO16		GPIO48	GPIOFS48.[0]	V	V	V	V	2-4mA
GPIO49	KSO17		GPIO49	GPIOFS48.[1]	V	V	V	V	2-4mA
GPIO4A	PSCLK1/ P80CLK		GPIO4A	GPIOFS48.[2]	V	V		V	2-4mA
GPIO4B	PSDAT1/ P80DAT		GPIO4B	GPIOFS48.[3]	V	V		V	2-4mA
GPIO4C	PSCLK2		GPIO4C	GPIOFS48.[4]	V	V		V	8-16mA
GPIO4D	PSDAT2		GPIO4D	GPIOFS48.[5]	V	V		V	8-16mA
GPIO4E	PSCLK3		GPIO4E	GPIOFS48.[6]	V	V		V	2-4mA
GPIO4F	PSDAT3		GPIO4F	GPIOFS48.[7]	V	V		V	2-4mA
GPIO50			GPIO50	GPIOFS50.[0]	V	V		V	2-4mA
GPIO51*			GPIO51	GPIOFS50.[1]					
GPIO52	E51CS#		GPIO52	GPIOFS50.[2]	V	V		V	8-16mA
GPIO53	CAPSLED#	E51TMR1	GPIO53	GPIOFS50.[3]	V	V		V	8-16mA
GPIO54	WDT_LED#	E51TMR0	GPIO54	GPIOFS50.[4]	V	V		V	8-16mA
GPIO55	SCORLED#	E51INT0	GPIO55	GPIOFS50.[5]	V	V		V	8-16mA
GPIO56		E51INT1	GPIO56	GPIOFS50.[6]	V	V	V	V	2-4mA
GPIO57	XCLK32K		GPIO57	GPIOFS50.[7]	V	V	V	V	2-4mA
GPIO58	SPICLK		GPIO58	GPIOFS58.[0]	V	V	V	V	8-16mA
GPIO59		TEST_CLK/ SPICLK	GPIO59	GPIOFS58.[1]	V	V	V	V	2-4mA
GPXIOA00	SDICS#			GPIO_MISC.[2]	V	V			2-4mA
GPXIOA01	SDICLK			GPIO_MISC.[2]	V	V			2-4mA
GPXIOA02	SDIDO			GPIO_MISC.[2]	V	V			2-4mA
GPXIOA03					V	V			2-4mA
GPXIOA04					V	V			2-4mA

GPIO	Alt. Output	Alt. Input	Default Alt. Output	Alt. Selection Reg.	Input Enable	Output Enable	Pull Up (40KΩ)	Open Drain	Output Current
GPXIOA05					V	V			2-4mA
GPXIOA06					V	V			2-4mA
GPXIOA07					V	V			2-4mA
GPXIOA08					V	V			8-16mA
GPXIOA09					V	V			8-16mA
GPXIOA10					V	V			8-16mA
GPXIOA11					V	V			8-16mA
GPXIOA12*									
GPXIOA13*									
GPXIOA14*									
GPXIOA15*									
GPXIOA16*									
GPXIOA17*									
GPXIOA18*									
GPXIOD00		SDIDI			V	V			2-4mA
GPXIOD01					V	V			2-4mA
GPXIOD02					V	V			2-4mA
GPXIOD03					V	V			2-4mA
GPXIOD04					V	V			2-4mA
GPXIOD05					V	V			2-4mA
GPXIOD06					V	V			2-4mA
GPXIOD07					V	V			2-4mA

* denotes that these pins do not exist in KB926D.

4.2.3 GPIO Registers Descriptions

Function Selection Register					
Offset	**Name**	**Type.**	**Description**	**Default**	**Bank**
0x00	GPIOFS00	R/W	GPIO00~GPIO07 Function Selection bit[0]~bit[7] stand for GPIO00~GPIO07 separately **0:** General purpose output function selected **1:** Alternative output function selected.	0x00	0xFC
0x01	GPIOFS08	R/W	GPIO08~GPIO0F Function Selection bit[0]~bit[7] stand for GPIO08~GPIO0F separately **0:** General purpose output function selected **1:** Alternative output function selected.	0x00	0xFC
0x02	GPIOFS10	R/W	GPIO10~GPIO17 Function Selection bit[0]~bit[7] stand for GPIO10~GPIO17 separately **0:** General purpose output function selected **1:** Alternative output function selected.	0x00	0xFC
0x03	GPIOFS18	R/W	GPIO18~GPIO1F Function Selection bit[0]~bit[7] stand for GPIO18~GPIO1F separately **0:** General purpose output function selected **1:** Alternative output function selected.	0x00	0xFC
0x04	GPIOFS20	R/W	GPIO20~GPIO27 Function Selection bit[0]~bit[7] stand for GPIO20~GPIO27 separately **0:** General purpose output function selected **1:** Alternative output function selected.	0x00	0xFC
0x05	GPIOFS28	R/W	GPIO28~GPIO2F Function Selection bit[0]~bit[7] stand for GPIO28~GPIO2F separately **0:** General purpose output function selected **1:** Alternative output function selected.	0x00	0xFC
0x06	GPIOFS30	R/W	GPIO30~GPIO37 Function Selection bit[0]~bit[7] stand for GPIO30~GPIO37 separately **0:** General purpose output function selected **1:** Alternative output function selected.	0x00	0xFC
0x07	GPIOFS38	R/W	GPIO3C~GPIO3F Function Selection bit[4]~bit[7] stand for GPIO3C~GPIO3F separately **0:** General purpose output function selected If DAC selected, please do not set these related bits to "1". *＊GPI38~GPI3B without alternative output function.*	0x00	0xFC
0x08	GPIOFS40	R/W	GPIO40~GPIO47 Function Selection bit[0]~bit[7] stand for GPIO40~GPIO47 separately **0:** General purpose output function selected **1:** Alternative output function selected. *＊GPI42~GPI43 without alternative output function.*	0x00	0xFC
0x09	GPIOFS48	R/W	GPIO48~GPIO4F Function Selection bit[0]~bit[7] stand for GPIO48~GPIO4F separately **0:** General purpose output function selected **1:** Alternative output function selected.	0x00	0xFC
0x0A	GPIOFS50	R/W	GPIO50~GPIO57 Function Selection bit[0]~bit[7] stand for GPIO50~GPIO57 separately **0:** General purpose output function selected **1:** Alternative output function selected.	0x02	0xFC
0x0B	GPIOFS58	R/W	GPIO58~GPIO59 Function Selection bit[0]~bit[1] stand for GPIO58~GPIO59 separately **0:** General purpose output function selected **1:** Alternative output function selected.	0x00	0xFC

Output Enable Register					
Offset	**Name**	**Type.**	**Description**	**Default**	**Bank**
0x10	GPIOOE00	R/W	GPIO00~GPIO07 Output Enable bit[0]~bit[7] stand for GPIO00~GPIO07 separately **0:** Output Disable **1:** Output Enable	0x00	0xFC
0x11	GPIOOE08	R/W	GPIO08~GPIO0F Output Enable bit[0]~bit[7] stand for GPIO08~GPIO0F separately **0:** Output Disable **1:** Output Enable	0x00	0xFC
0x12	GPIOOE10	R/W	GPIO10~GPIO17 Output Enable bit[0]~bit[7] stand for GPIO10~GPIO17 separately **0:** Output Disable **1:** Output Enable	0x00	0xFC
0x13	GPIOOE18	R/W	GPIO18~GPIO1F Output Enable bit[0]~bit[7] stand for GPIO18~GPIO1F separately **0:** Output Disable **1:** Output Enable	0x00	0xFC
0x14	GPIOOE20	R/W	GPIO20~GPIO27 Output Enable bit[0]~bit[7] stand for GPIO20~GPIO27 separately **0:** Output Disable **1:** Output Enable	0x00	0xFC
0x15	GPIOOE28	R/W	GPIO28~GPIO2F Output Enable bit[0]~bit[7] stand for GPIO28~GPIO2F separately **0:** Output Disable **1:** Output Enable	0x00	0xFC
0x16	GPIOOE30	R/W	GPIO30~GPIO37 Output Enable bit[0]~bit[7] stand for GPIO30~GPIO37 separately **0:** Output Disable **1:** Output Enable	0x00	0xFC
0x17	GPIOOE38	R/W	GPIO3C~GPIO3F Output Enable bit[4]~bit[7] stand for GPIO3C~GPIO3F separately **0:** Output Disable **1:** Output Enable ** GPI38~GPI3A without output enable feature.*	0x00	0xFC
0x18	GPIOOE40	R/W	GPIO40~GPIO47 Output Enable bit[0]~bit[7] stand for GPIO40~GPIO47 separately **0:** Output Disable **1:** Output Enable **GPI42~GPI43 without output enable.*	0x00	0xFC
0x19	GPIOOE48	R/W	GPIO48~GPIO4F Output Enable bit[0]~bit[7] stand for GPIO48~GPIO4F separately **0:** Output Disable **1:** Output Enable	0x00	0xFC
0x1A	GPIOOE50	R/W	GPIO50~GPIO57 Output Enable bit[0]~bit[7] stand for GPIO50~GPIO57 separately **0:** Output Disable **1:** Output Enable	0x02	0xFC

0x1B	GPIOOE58	R/W	GPIO58~GPIO59 Output Enable bit[0]~bit[1] stand for GPIO58~GPIO59 separately **0:** Output Disable **1:** Output Enable	0x00	0xFC
0x1C	GPXAOE00	R/W	GPXIOA00~GPXIOA07 Output Enable bit[0]~bit[7] stand for GPXIOA00~GPXIOA07 separately **0:** Output Disable **1:** Output Enable	0x00	0xFC
0x1D	GPXAOE08	R/W	GPXIOA08~GPXIOA15 Output Enable bit[0]~bit[7] stand for GPXIOA08~GPXIOA15 separately **0:** Output Disable **1:** Output Enable	0x00	0xFC
0x1E	GPXAOE16	R/W	GPXIOA16~GPXIOA18 Output Enable bit[0]~bit[2] stand for GPXIOA016~GPXIOA18 separately **0:** Output Disable **1:** Output Enable	0x00	0xFC
0x1F	GPXDOE00	R/W	GPXIOD00~GPXIOD07 Output Enable bit[0]~bit[7] stand for GPXIOD00~GPXIOD07 separately **0:** Output Disable **1:** Output Enable	0x00	0xFC

Output Data Port Register					
Offset	**Name**	**Type.**	**Description**	**Default**	**Bank**
0x20	GPIOD00	R/W	GPIO00~GPIO07 Output Data Port for output function. Bit[0]~bit[7] stand for GPIO00~GPIO07 separately	0x00	0xFC
0x21	GPIOD08	R/W	GPIO08~GPIO0F Output Data Port for output function. Bit[0]~bit[7] stand for GPIO08~GPIO0F separately	0x00	0xFC
0x22	GPIOD10	R/W	GPIO10~GPIO17 Output Data Port for output function. Bit[0]~bit[7] stand for GPIO10~GPIO17 separately	0x00	0xFC
0x23	GPIOD18	R/W	GPIO18~GPIO1F Output Data Port for output function. Bit[0]~bit[7] stand for GPIO18~GPIO1F separately	0x00	0xFC
0x24	GPIOD20	R/W	GPIO20~GPIO27 Output Data Port for output function. Bit[0]~bit[7] stand for GPIO20~GPIO27 separately	0x00	0xFC
0x25	GPIOD28	R/W	GPIO28~GPIO2F Output Data Port for output function. Bit[0]~bit[7] stand for GPIO28~GPIO2F separately	0x00	0xFC
0x26	GPIOD30	R/W	GPIO30~GPIO37 Output Data Port for output function. Bit[0]~bit[7] stand for GPIO30~GPIO37 separately	0x00	0xFC
0x27	GPIOD38	R/W	GPIO3C~GPIO3F Output Data Port for output function. Bit[4]~bit[7] stand for GPIO3C~GPIO3F separately ** GPI38~GPI3B have no output data ports.*	0x00	0xFC
0x28	GPIOD40	R/W	GPIO40~GPIO47 Output Data Port for output function. Bit[0]~bit[7] stand for GPIO40~GPIO47 separately	0x00	0xFC
0x29	GPIOD48	R/W	GPIO48~GPIO4F Output Data Port for output function. Bit[0]~bit[7] stand for GPIO48~GPIO4F separately	0x00	0xFC
0x2A	GPIOD50	R/W	GPIO50~GPIO57 Output Data Port for output function. Bit[0]~bit[7] stand for GPIO50~GPIO57 separately	0x00	0xFC
0x2B	GPIOD58	R/W	GPIO58~GPIO59 Output Data Port for output function. Bit[0]~bit[1] stand for GPIO58~GPIO59 separately	0x00	0xFC
0x2C	GPXAD00	R/W	GPXIOA00~GPXIOA07 Output Data Port for output function. Bit[0]~bit[7] stand for GPXIOA00~GPXIOA07 separately	0x00	0xFC
0x2D	GPXAD08	R/W	GPXIOA08~GPXIOA15 Output Data Port for output function. Bit[0]~bit[7] stand for GPXIOA08~GPXIOA15 separately	0x00	0xFC
0x2E	GPXAD16	R/W	GPXIOA16~GPXIOA18 Output Data Port for output function. Bit[0]~bit[2] stand for GPXIOA16~GPXIOA18 separately	0x00	0xFC
0x2F	GPXDD00	R/W	GPXIOD00~GPXIOD07 Output Data Port for output function. Bit[0]~bit[7] stand for GPXIOD00~GPXIOD07 separately	0x00	0xFC

Input Data Port Register					
Offset	**Name**	**Type.**	**Description**	**Default**	**Bank**
0x30	GPIOIN00	R	GPIO00~GPIO07 Input Data Port for input function. Bit[0]~bit[7] stand for GPIO00~GPIO07 separately	0xFF	0xFC
0x31	GPIOIN08	R	GPIO08~GPIO0F Input Data Port for input function. Bit[0]~bit[7] stand for GPIO08~GPIO0F separately	0xFF	0xFC
0x32	GPIOIN10	R	GPIO10~GPIO17 Input Data Port for input function. Bit[0]~bit[7] stand for GPIO10~GPIO17 separately	0xFF	0xFC
0x33	GPIOIN18	R	GPIO18~GPIO1F Input Data Port for input function. Bit[0]~bit[7] stand for GPIO18~GPIO1F separately	0xFF	0xFC
0x34	GPIOIN20	R	GPIO20~GPIO27 Input Data Port for input function. Bit[0]~bit[7] stand for GPIO20~GPIO27 separately	0xFF	0xFC
0x35	GPIOIN28	R	GPIO28~GPIO2F Input Data Port for input function. Bit[0]~bit[7] stand for GPIO28~GPIO2F separately	0xFF	0xFC
0x36	GPIOIN30	R	GPIO30~GPIO37 Input Data Port for input function. Bit[0]~bit[7] stand for GPIO30~GPIO37 separately	0xFF	0xFC
0x37	GPIOIN38	R	GPIO38~GPIO3B Input Data Port for input function. Bit[0]~bit[3] stand for GPIO38~GPIO3B separately ** GPO3C~GPO3F have no input data ports.*	0xFF	0xFC
0x38	GPIOIN40	R	GPIO40~GPIO47 Input Data Port for input function. Bit[0]~bit[7] stand for GPIO40~GPIO47 separately	0xFF	0xFC
0x39	GPIOIN48	R	GPIO48~GPIO4F Input Data Port for input function. Bit[0]~bit[7] stand for GPIO48~GPIO4F separately	0xFF	0xFC
0x3A	GPIOIN50	R	GPIO50~GPIO57 Input Data Port for input function. Bit[0]~bit[7] stand for GPIO50~GPIO57 separately	0xFF	0xFC
0x3B	GPIOIN58	R	GPIO58~GPIO59 Input Data Port for input function. Bit[0]~bit[1] stand for GPIO58~GPIO59 separately	0xFF	0xFC
0x3C	GPXAIN00	R	GPXIOA00~GPXIOA07 Input Data Port for input function. Bit[0]~bit[7] stand for GPXIOA00~GPXIOA07 separately	0xFF	0xFC
0x3D	GPXAIN08	R	GPXIOA08~GPXIOA15 Input Data Port for input function. Bit[0]~bit[7] stand for GPXIOA08~GPXIOA15 separately	0xFF	0xFC
0x3E	GPXAIN16	R	GPXIOA16~GPXIOA18 Input Data Port for input function. Bit[0]~bit[2] stand for GPXIOA16~GPXIOA18 separately	0xFF	0xFC
0x3F	GPXDIN00	R	GPXIOD00~GPXIOD07 Input Data Port for input function. Bit[0]~bit[7] stand for GPXIOD00~GPXIOD07 separately	0xFF	0xFC

Pull-up Enable Register					
Offset	**Name**	**Type.**	**Description**	**Default**	**Bank**
0x40	GPIOPU00	R/W	GPIO00~GPIO07 Internal Pull-Up Resistor Enable for input function bit[0]~bit[7] stand for GPIO00~GPIO07 separately **0:** Pull-Up resistor disable **1:** Pull-Up resistor enable	0x00	0xFC
0x41	GPIOPU08	R/W	GPIO08~GPIO0F Internal Pull-Up Resistor Enable for input function bit[0]~bit[7] stand for GPIO08~GPIO0F separately **0:** Pull-Up resistor disable **1:** Pull-Up resistor enable	0x08	0xFC
0x42	GPIOPU10	R/W	GPIO10~GPIO17 Internal Pull-Up Resistor Enable for input function bit[0]~bit[7] stand for GPIO10~GPIO17 separately **0:** Pull-Up resistor disable **1:** Pull-Up resistor enable	0x00	0xFC
0x43	GPIOPU18	R/W	GPIO18~GPIO1F Internal Pull-Up Resistor Enable for input function bit[0]~bit[7] stand for GPIO18~GPIO1F separately **0:** Pull-Up resistor disable **1:** Pull-Up resistor enable.	0x00	0xFC
0x44	GPIOPU20	R/W	GPIO20~GPIO27 Internal Pull-Up Resistor Enable for input function bit[0]~bit[7] stand for GPIO20~GPIO27 separately **0:** Pull-Up resistor disable **1:** Pull-Up resistor enable	0x0F	0xFC
0x45	GPIOPU28	R/W	GPIO28~GPIO2F Internal Pull-Up Resistor Enable for input function bit[0]~bit[7] stand for GPIO28~GPIO2F separately **0:** Pull-Up resistor disable **1:** Pull-Up resistor enable	0x00	0xFC
0x46	GPIOPU30	R/W	GPIO30~GPIO37 Internal Pull-Up Resistor Enable for input function bit[0]~bit[7] stand for GPIO30~GPIO37 separately **0:** Pull-Up resistor disable **1:** Pull-Up resistor enable	0xFF	0xFC
0x47	RSV	RSV	Reserved		0xFC
0x48	GPIOPU40	R/W	GPIO40~GPIO47 Internal Pull-Up Resistor Enable for input function bit[0]~bit[7] stand for GPIO40~GPIO47 separately **0:** Pull-Up resistor disable **1:** Pull-Up resistor enable	0x00	0xFC
0x49	GPIOPU48	R/W	GPIO48~GPIO4F Internal Pull-Up Resistor Enable for input function bit[0]~bit[7] stand for GPIO48~GPIO4F separately **0:** Pull-Up resistor disable **1:** Pull-Up resistor enable.	0x00	0xFC
0x4A	GPIOPU50	R/W	GPIO50~GPIO57 Internal Pull-Up Resistor Enable for input function bit[0]~bit[7] stand for GPIO50~GPIO57 separately **0:** Pull-Up resistor disable **1:** Pull-Up resistor enable	0x00	0xFC
0x4B	GPIOPU58	R/W	GPIO58~GPIO59 Internal Pull-Up Resistor Enable for input function bit[0]~bit[1] stand for GPIO58~GPIO59 separately **0:** Pull-Up resistor disable **1:** Pull-Up resistor enable.	0x00	0xFC

Open Drain Enable Register					
Offset	**Name**	**Type.**	**Description**	**Default**	**Bank**
0x50	GPIOOD00	R/W0C	GPIO00~GPIO07 Open Drain Enable for output function bit[0]~bit[7] stand for GPIO00~GPIO07 separately **0:** Open drain disable **1:** Open drain enable.	0x00	0xFC
0x51	GPIOOD08	R/W0C	GPIO08~GPIO0F Open Drain Enable for output function bit[0]~bit[7] stand for GPIO08~GPIO0F separately **0:** Open drain disable **1:** Open drain enable.	0x00	0xFC
0x52	GPIOOD10	R/W0C	GPIO10~GPIO17 Open Drain Enable for output function bit[0]~bit[7] stand for GPIO10~GPIO17 separately **0:** Open drain disable **1:** Open drain enable.	0x00	0xFC
0x53	GPIOOD18	R/W0C	GPIO18~GPIO1F Open Drain Enable for output function bit[0]~bit[7] stand for GPIO18~GPIO1F separately **0:** Open drain disable **1:** Open drain enable.	0x00	0xFC
0x54	GPIOOD20	R/W0C	GPIO20~GPIO27 Open Drain Enable for output function bit[0]~bit[7] stand for GPIO20~GPIO27 separately **0:** Open drain disable **1:** Open drain enable.	0x00	0xFC
0x55	GPIOOD28	R/W0C	GPIO28~GPIO2F Open Drain Enable for output function bit[0]~bit[7] stand for GPIO28~GPIO2F separately **0:** Open drain disable **1:** Open drain enable.	0x00	0xFC
0x56	GPIOOD30	R/W0C	GPIO30~GPIO37 Open Drain Enable for output function bit[0]~bit[7] stand for GPIO30~GPIO37 separately **0:** Open drain disable **1:** Open drain enable.	0x00	0xFC
0x57	RSV	RSV	RSV		0xFC
0x58	GPIOOD40	R/W0C	GPIO40~GPIO47 Open Drain Enable for output function bit[0]~bit[7] stand for GPIO40~GPIO47 separately **0:** Open drain disable **1:** Open drain enable.	0x00	0xFC
0x59	GPIOOD48	R/W0C	GPIO48~GPIO4F Open Drain Enable for output function bit[0]~bit[7] stand for GPIO48~GPIO4F separately **0:** Open drain disable **1:** Open drain enable.	0x00	0xFC
0x5A	GPIOOD50	R/W0C	GPIO50~GPIO57 Open Drain Enable for output function bit[0]~bit[7] stand for GPIO50~GPIO57 separately **0:** Open drain disable **1:** Open drain enable.	0x00	0xFC
0x5B	GPIOOD58	R/W0C	GPIO58~GPIO59 Open Drain Enable for output function bit[0]~bit[1] stand for GPIO58~GPIO59 separately **0:** Open drain disable **1:** Open drain enable.	0x00	0xFC

Input Enable Register					
Offset	**Name**	**Type.**	**Description**	**Default**	**Bank**
0x60	GPIOIE00	R/W	GPIO00~GPIO07 Input Enable for input function bit[0]~bit[7] stand for GPIO00~GPIO07 separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0x20	0xFC
0x61	GPIOIE08	R/W	GPIO08~GPIOF Input Enable for input function bit[0]~bit[7] stand for GPIO08~GPIO0F separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0x00	0xFC
0x62	GPIOIE10	R/W	GPIO10~GPIO17 Input Enable for input function bit[0]~bit[7] stand for GPIO10~GPIO17 separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0x00	0xFC
0x63	GPIOIE18	R/W	GPIO18~GPIO1F Input Enable for input function bit[0]~bit[7] stand for GPIO18~GPIO1F separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0x00	0xFC
0x64	GPIOIE20	R/W	GPIO20~GPIO27 Input Enable for input function bit[0]~bit[7] stand for GPIO20~GPIO27 separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0x0F	0xFC
0x65	GPIOIE28	R/W	GPIO28~GPIO2F Input Enable for input function bit[0]~bit[7] stand for GPIO28~GPIO2F separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0x00	0xFC
0x66	GPIOIE30	R/W	GPIO30~GPIO37 Input Enable for input function bit[0]~bit[7] stand for GPIO30~GPIO37 separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0xFF	0xFC
0x67	GPIOIE38	R/W	GPIO38~GPIO3B Input Enable for input function bit[0]~bit[3] stand for GPIO38~GPIO3B separately **0:** GPIO input mode disable **1:** GPIO input mode enable. ** GPO3C~GPO3F have no input functions.*	0x00	0xFC
0x68	GPIOIE40	R/W	GPIO40~GPIO47 Input Enable for input function bit[0]~bit[7] stand for GPIO40~GPIO47 separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0x00	0xFC
0x69	GPIOIE48	R/W	GPIO48~GPIO4F Input Enable for input function bit[0]~bit[7] stand for GPIO48~GPIO4F separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0x00	0xFC
0x6A	GPIOIE50	R/W	GPIO50~GPIO57 Input Enable for input function bit[0]~bit[7] stand for GPIO50~GPIO57 separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0x00	0xFC
0x6B	GPIOEE58	R/W	GPIO58~GPIO59 Input Enable for input function bit[0]~bit[1] stand for GPIO58~GPIO59 separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0x03	0xFC
0x6C	GPXAIE00	R/W	GPXIOA00~GPXIOA07 Input Enable for input function bit[0]~bit[7] stand for GPXIOA00~GPXIOA07 separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0x01	0xFC

0x6D	GPXAIE08	R/W	GPXIOA08~GPXIOA15 Input Enable for input function bit[0]~bit[7] stand for GPXIOA08~GPXIOA15 separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0x00	0xFC
0x6E	GPXAIE16	R/W	GPXIOA16~GPXIOA18 Input Enable for input function bit[0]~bit[2] stand for GPXIOA16~GPXIOA18 separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0x00	0xFC
0x6F	GPXDIE00	R/W	GPXIOD00~GPXIOD07 Input Enable for input function bit[0]~bit[7] stand for GPXIOD00~GPXIOD07 separately **0:** GPIO input mode disable **1:** GPIO input mode enable.	0x00	0xFC

GPIO_MISC Control Register

Offset	Name	Bit	Type	Description	Default	Bank
0x70	GPIO_MISC	7	R/W	ESB_DAT(GPIO0C) output current selection 0: 8mA 1: 16mA	0x0	0xFC
		6	R/W	SPICLK(GPIO58) output current selection 0: 8mA 1: 16mA		
		5	R/W	ESB_CLK(GPIO0B) output current selection 0: 8mA 1: 16mA		
		4	RSV	Reserved		
		3	R/W	GPIO17 / GPIO18 are featured with signal bypass function. Signal input via GPIO17 can be directly passed through GPIO18. **0:** Pass through function disable **1:** Pass through function enable		
		2	R/W	Alternative functions select for GPXIOA00~GPXIOA02. **0:** GPXIOA00~GPXIOA02 remain default output function **1:** GPXIOA00~GPXIOA02 become SDICS#, SDICLK, and SDIDO functions.		
		1	R/W	Debug port80 interface selection. **0:** Debug port80 interface using P80CLK/P80DAT **1:** Debug port80 interface using PSCLK1/PSDAT1 as P80CLK/P80DAT		
		0	R/W	Beep glue logic switch. GPIO12 can be output a specific function as following formula. *GPIO12 = PWM2 ⊕GPIO16(input) ⊕GPIO17(input)* **0:** Beep glue logic function disable **1:** Beep glue logic function enable		

4.2.4 GPIO Programming Sample

In this section gives some programming sample to control GPIO module. Please note, ENE does not guarantee these codes in every field application. The following table describes scenario of GPIO filed application.

Example	
PIN	**Function**
GPIO00 (GA20)	Output
GPIO01 (KBRST#)	Output
GPIO02 (GPIO)	Input
GPIO03 (GPIO)	Input
GPIO04 (GPIO)	Output
GPIO05 (PCIRST#)	Input
GPIO06 (GPIO)	Input
GPIO07 (GPIO)	Output
Programming model	
6. set function selection register. GPIOFS00 (0xFC00) = 0x03 2. set related pins to be output enable. GPIOOE00 (0xFC10) = 0x93 3. set related pins to be input enable. GPIOIE00 (0xFC60) = 0x6C	

4.3 Keyboard and Mouse Control Interface (KBC)

4.3.1 KBC I/F Function Description

The KBC is compatible with i8042 and responsible for keyboard/mouse accessing via legacy 60h/64h ports. The port 60h is the data port and port 64h is the command port. The legacy IRQ1 for keyboard devices and IRQ12 for mouse devices can be generated. The KBC interface provides fast GA20 control for legacy application.

KBC data register can be accessed by host or KBC firmware. Writing this register will setup a **OBF** (**O**utput **B**uffer **F**ull) flag, which can be clear by firmware. While the host issues I/O write to 60h/64h port, an **IBF** (**I**nput **B**uffer **F**ull) flag will assert. The interrupts can be programmed to issue while the flag of IBF/OBF asserting.

The following table gives a summary about port 60h/64h accessing.

Port	Access	Type	Register	Flag	Comment
60h	I/O Write	Data	KBCDAT (0xFC85)	IBF	Write data to keyboard/mouse
64h	I/O Write	Command	KBCCMD (0xFC84)	IBF	Write command to keyboard/mouse
60h	I/O Read	Data	KBCDAT (0xFC85)	OBF	Read data from keyboard/mouse
64h	I/O Read	Status	KBCSTS (0xFC86)		Read status from keyboard/mouse

KBC data register, **KBCDAT**, keeps data from host or data written by KBC firmware.

Bit	7	6	5	4	3	2	1	0
Name	Keyboard/Mouse Data Register							

KBC command register, **KBCCMD**, is used to keep the command from host. This register is read only.

Bit	7	6	5	4	3	2	1	0
Name	Keyboard/Mouse Command Register							

KBC status register, **KBCSTS**, keeps the status as the following table. For more detail please refer to the section, **KBC Registers Description**.

Bit	7	6	5	4	3	2	1	0
Name	Parity Error	Time Out	Aux. Data Flag	Uninhibited	Address (A2)	System Flag	IBF	OBF

4.3.2 KBC Registers Description

KBC Command Byte Register (KBC command 20h/60h)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x80	KBCCB	7	R/W	PS/2 hardware mode enable. **0**: Disable **1**: Enable If the host issues command 20h via port 64h, and the KBC returns data via port 60h. This bit will always be read as **zero**.	0x40	0xFC
		6	R/W	Scan code set2 conversion enable (PS/2 scan code set2 converts to set 1) **0**: Disable **1**: Enable		
		5	R/W	Disable Auxiliary device **0**: Enable **1**: Disable		
		4	R/W	Disable Keyboard device **0**: Enable **1**: Disable		
		3	R/W	Inhibit Override **0**: Disable **1**: Enable		
		2	R/W	System Flag (warm boot flag) **0**: cold boot **1**: warm boot		
		1	R/W	IRQ12 Enable While KBCSTS[5]=1(Auxiliary Data Flag) and KBCSTS[0]=1 (OBF), then IRQ12 will issue. **0**: Disable **1**: Enable		
		0	R/W	IRQ1 Enable While KBCSTS[5]=0 (Auxiliary Data Flag) and KBCSTS[0]=1 (OBF), then IRQ1 will issue. **0**: Disable **1**: Enable		

KBC Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x81	KBCCFG	7	R/W	Keyboard lock enable **0**: Disable **1**: Enable	0x00	0xFC
		6	R/W	Fast gate A20 control **0**: Disable gate A20 control **1**: Enable gate A20 control		
		5	R/W	KBC hardware command sets (90h~93h, D4h) enable. **0**: Disable **1**: Enable		
		4	R/W	KBC hardware command sets (60h, A7h~ABh, Adh~Aeh) enable. **0**: Disable **1**: Enable		
		3	R/W	Keyboard lock flag status **0**: keyboard not lock or not inhibit **1**: keyboard lock or inhibit		
		2	R/W	KBC hardware command sets (A4h, A6h) enable. **0**: Disable **1**: Enable		
		1	R/W	IBF (KBCSTS[1]) interrupt enable. (IBF from 0 to 1) **0**: Disable **1**: Enable		
		0	R/W	OBF (KBCSTS[0]) interrupt enable (OBF from 1 to 0) **0**: Disable **1**: Enable		

KBC Interrupt Pending Flag						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x82	KBCIF	7-3	RSV	Reserved	0x00	0xFC
		2	R/W1C	Status of KBC command handled by firmware While receiving KBC commands which need firmware to handle, the hardware will set this bit. Then the firmware will deal with all the following command until this bit is clear by firmware.		
		1	R/W1C	IBF interrupt pending flag **0**: no IBF interrupt occurs **1**: IBF interrupt occurs		
		0	R/W1C	OBF interrupt pending flag **0**: no OBF interrupt occurs **1**: OBF interrupt occurs		

KBC Hardware Command Enable						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x83	KBCHWEN	7	R/W	KBC hardware command set (Feh) enable **0**: Disable **1**: Enable	0x00	0xFC
		6	R/W	KBC hardware command set (E0h) enable **0**: Disable **1**: Enable		
		5	R/W	KBC hardware command set (D3h) enable **0**: Disable **1**: Enable		
		4	R/W	KBC hardware command set (D2h) enable **0**: Disable **1**: Enable		
		3	R/W	KBC hardware command set (D1h) enable **0**: Disable **1**: Enable		
		2	R/W	KBC hardware command set (D0h) enable **0**: Disable **1**: Enable		
		1	R/W	KBC hardware command set (C0h) enable **0**: Disable **1**: Enable		
		0	R/W	KBC hardware command set (20h) enable **0**: Disable **1**: Enable		

KBC Command Buffer						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x84	KBCCMD	7-0	RO	Command written to port 64h will be stored in this register	0x00	0xFC

KBC Data Input/Output Buffer						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x85	KBCDAT	7-0	R/W	Data written to this register to make OBF set (OBF=1). The host read this register via port 60h.	0x00	0xFC

KBC Host Status						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x86	KBCSTS	7	R/W	Parity error **0**: No parity error occurs in PS/2 protocol **1**: Parity error occurs in PS/2 protocol.	0x00	0xFC
		6	R/W	Timeout **0**: No timeout occurs in PS/2 protocol **1**: Timeout occurs in PS/2 protocol.		
		5	R/W	Auxiliary data flag		
		4	RO	Uninhibited **0**: keyboard inhibited **1**: keyboard not inhibited		
		3	RO	Address (A2) **0**: output buffer data from 60h **1**: output buffer data from 64h		
		2	RO	System flag		
		1	R/W1C	IBF		
		0	R/W1C	OBF		

Reserved						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x88	RSV	7-0	RSV	Reserved	0x00	0xFC

Reserved						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x88	RSV	7-0	RSV	Reserved	0x00	0xFC

Reserved						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x89	RSV	7-0	RSV	Reserved	0x00	0xFC

KBC Write Data						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x8A	KBCDATR	7-0	RO	Read back port of KBCDAT, 0xFC85	0x00	0xFC

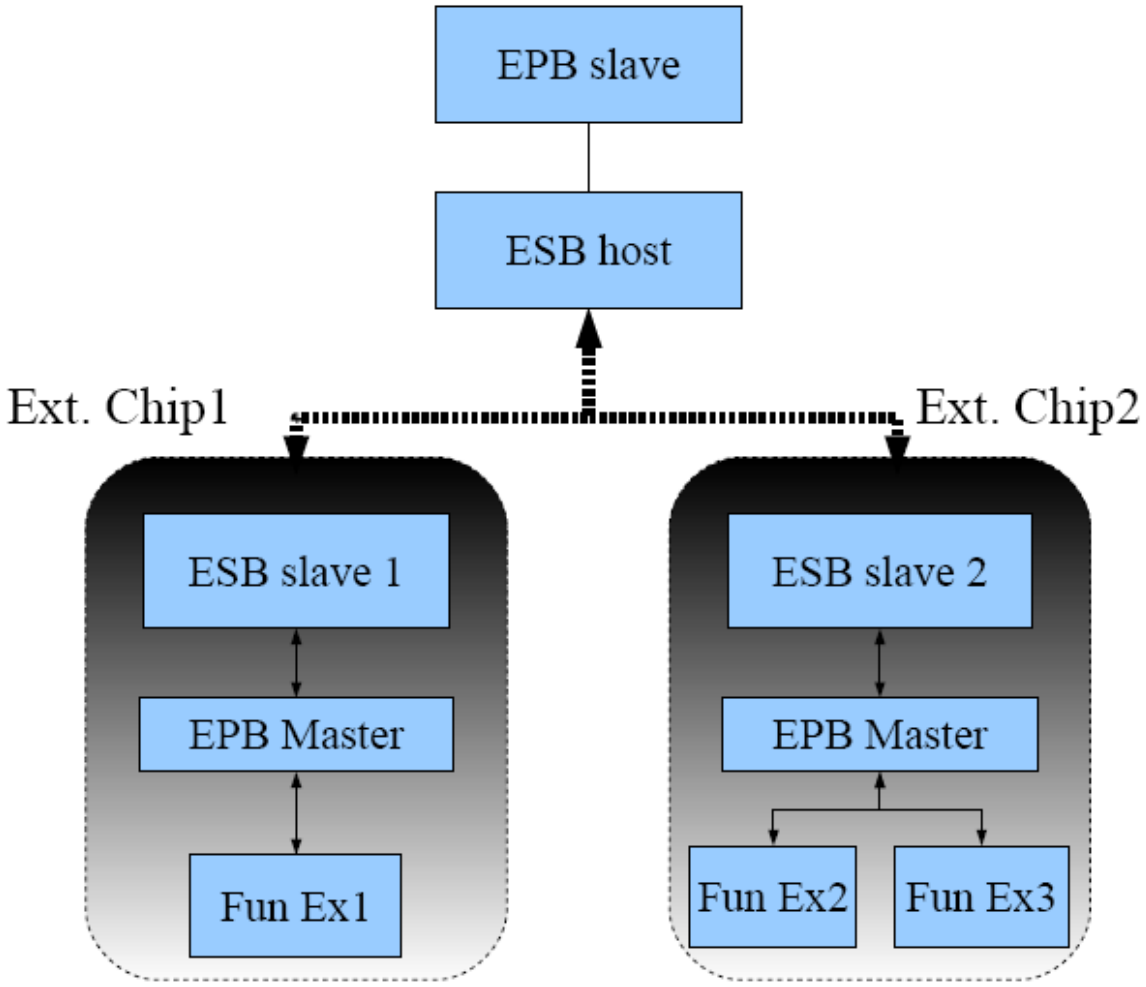
4.4 ENE Serial Bus Controller (ESB)

4.4.1 ESB Function Description

To extend the usage of the current design, an ENE serial bus interface is introduced. An external ESB device can be controlled by firmware transparently. As the following table, 4 memory address ranges are reserved for ESB devices.

	Memory Range
Range1	0xFCA0~0xFCAF
Range2	0xFCB0~0xFCBF
Range3	0xFCC0~0xFCCF
Range4	0xFD00~0xFDFF

In the ESB architecture, external ESB devices are supported. And each device can be configured with interrupt capability. A figure gives the topology of ENE Serial Bus as following.



The topology of ENE Serial Bus

4.4.2 ESB Registers Description

ESB Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x90	ESBCFG	7	R/W	Loop back test enable **0**: Disable **1**: Enable	0x00	0xFC
		6-5	R/W	ESB clock selection. **00**: (main clock) / 8 **01**: (main clock) / 4 **10**: (main clock) / 2 **11**: (main clock) / 1		
		4	R/W	External device access mode. **0**: Access external device via 4 predefined memory ranges. (automatic mode) **1**: Access external devices via **ESBCA**, **ESBCD** and **ESBRD** registers. (byte mode)		
		3	R/W	ESB clock output enable **0**: Disable **1**: Enable		
		2	R/W	ESB interrupt enable **0**: Disable **1**: Enable		
		1	R/W	ESB host queries device interrupt status automatically. (when **ESBCFG**[3]=1) **0**: Disable **1**: Enable		
		0	R/W	ESB function enable **0**: Disable **1**: Enable		

ESB Command and Status						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x91	ESBCS	7	RSV	Reserved	0x00	0xFC
		6	R/W1C	Device resume signal flag. **0**: no event **1**: event occurs.		
		5	R/W1C	ESB bus timeout status **0**: no timeout event **1**: bus timeout		
		4	R/W1C	Device data received status. **0**: no data received **1**: data received.		
		3	R	ESB host busy flag. **0**: not busy **1**: host busy		
		2	W	Start to send command, command byte in **ESBCD**, 0xFC94 Please write "0" will not work. **1**: send command		
		1-0	R/W	ESB access command type (while **ESBCFG**[3]=1) **00**: interrupt query **01**: read **10**: write **11**: Reserved		

ESB Interrupt Enable of External Device						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x92	ESBINTE	7	RSV	Reserved	0x00	0xFC
		6	R/W	Device resume signal interrupt enable **0**: Disable **1**: Enable		
		5	R/W	Bus timeout interrupt enable **0**: Disable **1**: Enable		
		4	R/W	Device data received interrupt enable **0**: Disable **1**: Enable		
		3	R/W	Interrupt enable (IRQ3) of external ESB device. **0**: Disable **1**: Enable		
		2	R/W	Interrupt enable (IRQ2) of external ESB device. **0**: Disable **1**: Enable		
		1	R/W	Interrupt enable (IRQ1) of external ESB device. **0**: Disable **1**: Enable		
		0	R/W	Interrupt enable (IRQ0) of external ESB device. **0**: Disable **1**: Enable		

ESB Command Address						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x93	ESBCA	7-0	R/W	External ESB device address to be accessed. (when **ESBCFG**[3]=1) The address is predefined according to different device.	0x00	0xFC

ESB Command Data						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x94	ESBCD	7-0	R/W	Write data port to external ESB device (when **ESBCFG**[3]=1)	0x00	0xFC

ESB Received Data						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x95	ESBRD	7-0	R/W	Read data port to external ESB device (when **ESBCFG**[3]=1) If loop back test enabled, ESBCFG[7]=1, the register will be writable, otherwise, read-only.	0x00	0xFC

ESB Enable for External Device						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x96	ESBED	7-5	RSV	Reserved	0x00	0xFC
		4	R/W	Low clock mode enable (clock source 32KHz) For performance and power saving consideration, while low clock mode enabled, please set the query function off. **0**: Disable **1**: Enable		
		3	R/W	Enable external ESB device decoding address 0xFEE0~0xFEFF **0**: Disable **1**: Enable		
		2	R/W	Enable external ESB device decoding address 0xFCC0~0xFCCF **0**: Disable **1**: Enable		
		1	R/W	Enable external ESB device decoding address 0xFCB0~0xFCBF **0**: Disable **1**: Enable		
		0	R/W	Enable external ESB device decoding address 0xFD00~0xFDFF. **0**: Disable **1**: Enable		

ESB Interrupt Event Pending Flag for External Chip						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x97	ESBINT	7	R/W1C	Interrupt event pending flag of IRQ7 (cascade mode only) **0**: no event **1**: event occurs	0x00	0xFC
		6	R/W1C	Interrupt event pending flag of IRQ6 (cascade mode only) **0**: no event **1**: event occurs		
		5	R/W1C	I Interrupt event pending flag of IRQ5 (cascade mode only) **0**: no event **1**: event occurs		
		4	R/W1C	Interrupt event pending flag of IRQ4 (cascade mode only) **0**: no event **1**: event occurs		
		3	R/W1C	Interrupt event pending flag of IRQ3 **0**: no event **1**: event occurs		
		2	R/W1C	Interrupt event pending flag of IRQ2 **0**: no event **1**: event occurs		
		1	R/W1C	Interrupt event pending flag of IRQ1 **0**: no event **1**: event occurs		
		0	R/W1C	Interrupt event pending flag of IRQ0 **0**: no event **1**: event occurs		

ESB Cascade Mode Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x98	ESBCAS	7	R/W	Interrupt enable of IRQ7 for external chip **0**: disable **1**: enable	0x00	0xFC
		6	R/W	Interrupt enable of IRQ6 for external chip **0**: disable **1**: enable		
		5	R/W	Interrupt enable of IRQ5 for external chip **0**: disable **1**: enable		
		4	R/W	Interrupt enable of IRQ4 for external chip **0**: disable **1**: enable		
		3-1	RSV	Reserved		
		0	R/W	Cascade mode enable **0**: disable **1**: enable		

4.4.3 ESB Programming Sample

In this section gives some programming sample to control ESB module. Please note, ENE does not guarantee these codes in every field application. The following table describes scenario of ESB filed application.

Example
A device connecting to ESB master.
Programming model
GPIOFS08[4:3] (0xFC01[4:3])= 11b ; ESB function selection pin GPIOIE08[4] (0xFC61[4]) = 1b ; set related pin as an input ESBCFG (0xFC90) = 0x69 ; ESB clock=32MHz / EPB mode enable ESBED (0xFC96) = 0x02 ; enable ESB Now F/W can access register 0xFCC0~0xFCCF

4.5 Internal KeyBoard (IKB) Encoder

4.5.1 IKB Function Description

The KBC supports internal keyboard encoder (IKB) in the notebook system. Here is the feature highlight of IKB module.

- Support 18x8 matrix.
- Keyboard scan output (KSO) 18 lines.
- Keyboard scan input (KSI) 8 lines
- KSO/KSI can be programmed to be GPIO
- KSO/KSI internal programmable pull-high feature supported.

4.5.2 IKB Registers Description

IKB Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA0	IKBCFG	7	R/W	IKB scan controller test mode enable. **0**: Disable **1**: Enable	0x00	0xFC
		6	R/W	IKB PS/2 wait time setting. The IKB makes sure PS/2 bus idle for specific time and then transmit the scan codes. **0**: 8 μs **1**: 64 μs		
		5	RSV	Reserved		
		4	WO	Force controller to scan key matrix. Write "**1**" to start.		
		3	RSV	Reserved		
		2	R/W	IKB scan repeat enable. Set this bit force the IKB controller to scan every 30ms. **0**: Disable **1**: Enable		
		1	R/W	Standard KB command hardware mode enable. Once the IKB received standard KB command, the hardware will handle it. **0**: Disable **1**: Enable		
		0	R/W	IKB scan controller enable. **0**: Disable **1**: Enable		

IKB LED Control						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA1	IKBLED	7	R/W	NumLock key **0**: Fn-Lock **1**: NumLock =Fn-Lock	0x00	0xFC
		6	R/W	Flag of Fn-Shift (in hardware mode) **0**: Fn-Shift not pressed **1**: Fn-Shift pressed		
		5	R/W	Flag of Fn-Lock (in hardware mode) **0**: Fn-Lock not pressed **1**: Fn-Lock pressed		
		4	R/W	LED output polarity, CapLock/NumLock/ScrLock output **0**: positive logic **1**: negative logic		
		3	RSV	Reserved		
		2	R/W	CapLock LED driving Once EC firmware gets the command to light up CapLock LED, F/W will setup this bit.		
		1	R/W	NumLock LED driving Once EC firmware gets the command to light up NumLock LED, F/W will setup this bit.		
		0	R/W	ScrLock LED driving Once EC firmware gets the command to light up ScrLock LED, F/W will setup this bit.		

IKB Typematic Control						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA2	IKBTYPEC	7	RSV	Reserved	0x00	0xFC
		6-5	R/W	1st key repeat delay time selection. **00b**: 250ms **01b**: 500ms **10b**: 750ms **11b**: 1 sec		
		4-0	R/W	Typematic repeat characters per second. **1Fh**: 2 char/sec **10h**: 10 char/sec **1Bh**: 3 char/sec **0Dh**: 12 char/sec **18h**: 4 char/sec **0Bh**: 15 char/sec **17h**: 5 char/sec **08h**: 16 char/sec **15h**: 6 char/sec **05h**: 20 char/sec **13h**: 8 char/sec **00h**: 30 char/sec		

IKB Interrupt Enable						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA3	IKBIE	7-6	RSV	Reserved	0x00	0xFC
		5	R/W	Interrupt enable. While the following commands handled by hardware occur. KB reset / KB disable / Non-standard hardware mode command **0**: Disable **1**: Enable		
		4	R/W	IKB RX finished interrupt enable. **0**: Disable **1**: Enable		
		3	R/W	IKB TX finished interrupt enable. **0**: Disable **1**: Enable		
		2	R/W	IKB typmatic repeat timeout interrupt enable. **0**: Disable **1**: Enable		
		1	R/W	IKB scan code finished interrupt enable. (IKBHCFG[0]=0) **0**: Disable **1**: Enable Or IKB break key (hotkey) interrupt enable. (IKBHCFG[0]=1) **0**: Disable **1**: Enable		
		0	R/W	IKB make key interrupt enable. (IKBHCFG[0]=0) **0**: Disable **1**: Enable Or IKB make key (hotkey) interrupt enable. (IKBHCFG[0]=1) **0**: Disable **1**: Enable		

IKB Interrupt Pending Flag						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA4	IKBPF	7	WO	Force the IKB controller enter idle mode. Write "**1**" to enter idle mode.	0x00	0xFC
		6	R/W1C	**IKBSADR** (0xFCA9) valid flag. **0**: no more valid IKBSADR **1**: IKBSADR valid		
		5	R/W1C	Interrupt flag. While the following commands handled by hardware occur. KB reset / KB disable **0**: event is not active **1**: event is active		
		4	R/W1C	IKB RX finished and non-standard hardware mode command occurring interrupt flag. **0**: event is not active **1**: event is active		
		3	R/W1C	IKB TX finished interrupt flag. **0**: event is not active **1**: event is active		
		2	R/W1C	IKB typmatic repeat timeout interrupt flag **0**: event is not active **1**: event is active		
		1	R/W1C	IKB scan code finished interrupt flag. (IKBHCFG[0]=0) IKB break key (hotkey) interrupt flag. (IKBHCFG[0]=1) **0**: event is not active **1**: event is active		
		0	R/W1C	IKB make key interrupt flag. (IKBHCFG[0]=0) IKB make key (hotkey) interrupt flag. (IKBHCFG[0]=1) **0**: Disable **1**: Enable		

IKB PS/2 TX Data Byte						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA5	IKBTXDAT	7-0	R/W	The IKB port to transmit data to PS/2 controller Writing to this port, the data will be delivered to PS/2 controller. After transmission completes and a TX finished interrupt issues.	0x00	0xFC

IKB PS/2 RX Data Byte						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA6	IKBRXDAT	7-0	R/W	The IKB port to receive data from PS/2 controller. After receiving data from PS/2 controller, a RX finished interrupt issues.	0x00	0xFC

IKB Hardware Mode Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA7	IKBHCFG	7-3	RSV	Reserved	0x00	0xFC
		2	R/W	IKB hotkey flag while hardware mode enable (IKBCFG[0]=1) **0**: event is not active **1**: event is active		
		1	R/W	IKB hotkey finish indicator While KBC recognizes a hotkey, the KBC setup the hotkey flag (IKBCFG[2]) to invoke firmware to handle. Firmware will write "**1**" to this bit after completing the hotkey event.		
		0	R/W	IKB hardware mode enable **0**: Disable **1**: Enable		

IKB Scan Inputs						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA8	IKBKSI	7-0	RO	IKB scan input buffer	0x00	0xFC

IKB Scan Address						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA9	IKBSADR	7-0	RO	IKB scan address of current key	0x00	0xFC

IKB Scan Timing Control						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xAA	IKBSDB	7-4	R/W	KSO release (floating) time Time = (value + 1) * 8 μs	0xF7	0xFC
		3-0	R/W	KSO drive low time Time = (value + 1) * 8 μs		

IKB Make Key (hardware mode)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xAB	IKBMK	7-0	RO	The scan controller places make key in this register. If hotkey occurs, the register contains the matrix value.	0x00	0xFC

IKB Break Key (hardware mode)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xAC	IKBBK	7-0	RO	The scan controller places break key in this register. If hotkey occurs, the register contains the matrix value.	0x00	0xFC

Reserved						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xAD	RSV	7-0	RSV	Reserved	0x00	0xFC

IKB Matrix Table Address (hardware mode)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xAE	IKBMTA	7-3	RSV	RSV	0x00	0xFC
		2-0	R/W	IKB Matrix Table Address = 0xF400 + (IKBMTA * 0x100) **0**: 0xF400~0xF4FF **4**: 0xF800~0xF8FF **1**: 0xF500~0xF5FF **5**: 0xF900~0xF9FF **2**: 0xF600~0xF6FF **6**: 0xFA00~0xFAFF **3**: 0xF700~0xF7FF **7**: 0xFB00~0xFBFF		

IKB Key Generation Flag						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xAF	IKBKGENFG	7-3	RSV		0x00	0xFC
		2	RO	IKB code buffer full flag. When the code buffer full, this flag will be set. **0**: not over 8 keys in the code buffer. **1**: 8 keys in the code buffer..		
		1	R/W1C	Ghost key identification flag (IKBHCFG[0]=1) **0**: No ghost key **1**: Ghost key found		
		0	R/W1C	IKB make key scan flag. If this bit is set to “1”, all the make keys will be ignored. **0**: not over 5 make key occur at a time **1**: over 5 make key occur at a time		

4.5.3 IKB Matrix Value Mapping Table

In this section, the following tables show the mapping information between matrix value and PS/2 set1 scan code. The first one is the standard keys mapping, and the second one is for multimedia keys mapping.

Standard Keys					
Matrix Value (set 2)	**Description**	**Scan Code (set 1)**	**Matrix Value (set 2)**	**Description**	**Scan Code (set 1)**
00h	Error(overrun)	FFh	40h	Reserved	6Bh
01h	F9	43h	41h	< ,	33h
02h	F7	41h	42h	K	25h
03h	F5	3Fh	43h	I	17h
04h	F3	3Dh	44h	O	18h
05h	F1	3Bh	45h	) 0	0Bh
06h	F2	3Ch	46h	(9	0Ah
07h	F12	58h	47h	Reserved	60h
08h	Reserved	64h	48h	Reserved	6Ch
09h	F10	44h	49h	> .	34h
0Ah	F8	42h	4Ah	? /	35h
0Bh	F6	40h	4Bh	L	26h
0Ch	F4	3Eh	4Ch	: ;	27h
0Dh	Tab	0Fh	4Dh	P	19h
0Eh	~	29h	4Eh	_ -	0Ch
0Fh	Reserved	59h	4Fh	Reserved	61h
10h	Reserved	65h	50h	Reserved	6Dh
11h	Left Alt	38h	51h	Reserved	73h
12h	Left Shift	2Ah	52h	" '	28h
13h	Reserved	70h	53h	Reserved	74h
14h	Left Ctrl	1Dh	54h	{ [	1Ah
15h	Q	10h	55h	+ =	0Dh
16h	! 1	02h	56h	Reserved	62h
17h	Reserved	5Ah	57h	Reserved	6Eh
18h	Reserved	66h	58h	Caps Lock	3Ah
19h	Reserved	71h	59h	Right Shift	36h
1Ah	Z	2Ch	5Ah	Return	1Ch
1Bh	S	1Fh	5Bh	}]	1Bh
1Ch	A	1Eh	5Ch	Reserved	75h
1Dh	W	11h	5Dh	|\(US only) ~#(102-key)	2Bh
1Eh	@ 2	03h	5Eh	Reserved	63h
1Fh	Reserved	5Bh	5Fh	Reserved	76h
20h	Reserved	67h	60h	Fn (PTL)	55h
21h	C	2Eh	61h	|\(102-key)	56h
22h	X	2Dh	62h	Reserved	77h

23h	D	20h	63h	Reserved	78h
24h	E	12h	64h	Reserved	79h
25h	$ 4	5Ch	65h	Reserved	7Ah
26h	# 3	04h	66h	Backspace	0Eh
27h	Reserved	05h	67h	Reserved	7Bh
28h	Reserved	68h	68h	Reserved	7Ch
29h	Space	39h	69h	1 End	4Fh
2Ah	V	2Fh	6Ah	Reserved	7Dh
2Bh	F	21h	6Bh	4 Left Arrow	4Bh
2Ch	T	14h	6Ch	7 Home	47h
2Dh	R	13h	6Dh	Reserved	7Eh
2Eh	% 5	06h	6Eh	Reserved	7Fh
2Fh	Reserved	5Dh	6Fh	Reserved	6Fh
30h	Reserved	69h	70h	0 Ins	52h
31h	N	31h	71h	. Del	53h
32h	B	30h	72h	2 Down Arrow	50h
33h	H	23h	73h	5	4Ch
34h	G	22h	74h	6 Right Arrow	4Dh
35h	Y	15h	75h	8 Up Arrow	48h
36h	^ 6	07h	76h	ESC	01h
37h	Reserved	5Eh	77h	Num Lock	45h
38h	Reserved	6Ah	78h	F11	57h
39h	Reserved	72h	79h	+	4Eh
3Ah	M	32h	7Ah	3 PgDn	51h
3Bh	J	24h	7Bh	-	4Ah
3Ch	U	16h	7Ch	*	37h
3Dh	& 7	08h	7Dh	9 PgUp	49h
3Eh	* 8	09h	7Eh	Scroll Lock	46h
3Fh	Reserved	5Fh	7Fh	Sys Req (84-key)	54h

Multimedia Keys					
Matrix Value (set 2)	**Description**	**Scan Code (set 1)**	**Matrix Value (set 2)**	**Description**	**Scan Code (set 1)**
00h – 7Fh	Standard Keys	See table above	9Ah	ACPI Sleep	E0 5F
80h	Left Shift	2Ah	9Bh	ACPI Wake	E0 63
81h	Left Ctrl	1Dh	9Ch	Left Window	E0 5B
82h	Left Alt	38h	9Dh	Right Window	E0 5C
83h	F7	41h	9Eh	Windows App	E0 5D
84h	SysReq	54h	9Fh	Break	1D E0 46
85h	Right Shift	36h	A0h	Volume Up	E0h 30h
86h	Right Ctrl	E0h 1Dh	A1h	Volume Down	E0h 2Eh
87h	Right Alt	E0h 38h	A2h	Next	E0h 19h
88h	Print Screen	E0h 2Ah E0h 37h	A3h	Previous	E0h 10h
89h	Pause	E1h 1Dh 45h	A4h	Stop	E0h 24h
8Ah	Insert	E0h 52h	A5h	Play/Pause	E0h 22h
8Bh	Home	E0h 47h	A6h	Mute	E0h 20h
8Ch	Page Up	E0h 49h	A7h	Media Select	E0h 6Dh
8Dh	Delete	E0h 53h	A8h	Email Reader	E0h 6Ch
8Eh	End	E0h 4Fh	A9h	Calculator	E0h 21h
8Fh	Page Down	E0h 51h	Aah	My Computer	E0h 6Bh
90h	Up Arrow	E0h 48h	Abh	WWW Search	E0h 65h
91h	Left Arrow	E0h 41h	Ach	WWW Home	E0h 32h
92h	Down Arrow	E0h 50h	Adh	WWW Back	E0h 6Ah
93h	Right Arrow	E0h 4Dh	Aeh	WWW Forward	E0h 69h
94h	/	E0h 35h	Afh	WWW Stop	E0h 68h
95h	Enter	E0h 1Ch	B0h	WWW Refresh	E0h 67h
96h	Fn Shift	No scan code	B1h	WWW Favor	E0h 66h
97h	Fn Lock	No scan code	B2h	OADG	45h/46h
98h	Num/Fn Lock	45h	B3h	Empty Key	No scan code
99h	ACPI Power	E0h 5Eh	B4h – FFh	Hot Key	

4.6 Pulse Width Modulation (PWM)

4.6.1 PWM Function Description

Pulse width modulation (PWM) is a powerful technique for controlling analog circuits with a processor's digital outputs. PWM is employed in a wide variety of applications, ranging from measurement and communications to power control and conversion.

The KBC supports 4 PWM channels. 2 channels (PWM0/PWM1) are for 8-bit resolution and 2 channels (PWM2/PWM3) are for 14-bit resolution. The PWM provides clock source selection which is defined in the register description.

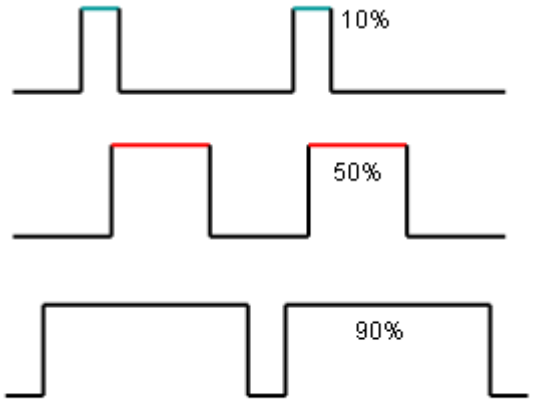


Figure. PWM Duty Cycle

. The duty cycle of PWM can be illustrated as the above figure. The following summarizes the relationship about the applications with the definition in the PWM registers description.

Definition	Formula	Comment
Duty Cycle	(PWM High Period Length+1)/(PWM Cycle Period Length+1) *100%	
Cycle Length	(PWM Cycle Length Register +1) * (PWM clock source)	For 8-bit
Cycle Length	(PWMCYC + 1) * 2 * (1 + Prescaler)/(Peripheral clock or fixed 1 MHz)	For 14-bit

For the limitation of current design, in some critical cases, the PWM output will be the one as the following table.

Condition	PWM Output
H>C	Always "1" (High)
H=0x00 and C=0x00	Always "1" (High)
H=0x00 and C=0xFF	A Short Pulse
H=0xFF and C=0x00	Always "1" (High)
Switch to GPIO mode and output low	Always "0" (Low)
H= High Period Length (PWMHIGH) , **C**= Cycle Period Length (PWMCYCL)	

4.6.2 PWM Registers Description

PWM Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x00	PWMCFG	7-6	R/W	PWM1 clock source selection **0**: 0.976 μs (1 μs) **1**: 62.5 μs (64 μs) **2**: 250 μs (2561 μs) **3**: 3.99ms (4ms)	0x00	0xFE
		5	RSV	Reserved		
		4	R/W	PWM1 Enable **0**: Disable **1**: Enable		
		3-2	R/W	PWM0 clock source selection **0**: 0.976 μs (1 μs) **1**: 62.5 μs (64 μs) **2**: 250 μs (2561 μs) **3**: 3.99ms (4ms)		
		1	RSV	Reserved		
		0	R/W	PWM0 Enable **0**: Disable **1**: Enable		

PWM0 High Period Length						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x01	PWMHIGH0	7-0	R/W	High Period Length of PWM0. This should be smaller than Cycle Length.	0x00	0xFE

PWM0 Cycle Length						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x02	PWMCYC0	7-0	R/W	Cycle Length of PWM0.	0x00	0xFE

PWM1 High Period Length						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x03	PWMHIGH1	7-0	R/W	High Period Length of PWM1. This should be smaller than Cycle Length.	0x00	0xFE

PWM1 Cycle Length						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x04	PWMCYC1	7-0	R/W	Cycle Length of PWM1	0x00	0xFE

Reserved						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x05	RSV	7-0	RSV	RSV	0x00	0xFE

PWM2 Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x06	PWMCFG2	7	R/W	PWM2 Enable **0**: Disable **1**: Enable	0x00	0xFE
		6	R/W	PWM2 prescaler clock selection **0**: peripheral clock **1**: 1MHz clock (fixed)		
		5-0	R/W	The 6-bit prescaler of PWM2 The precaler value = register value + 1		

PWM3 Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x07	PWMCFG3	7	R/W	PWM3 Enable **0**: Disable **1**: Enable	0x00	0xFE
		6	R/W	PWM3 prescaler clock selection **0**: peripheral clock **1**: 1MHz clock (fixed)		
		5-0	R/W	The 6-bit prescaler of PWM3 The precaler value = register value + 1		

PWM2 High Period Length (14-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x08	PWMHIGH2H	5-0	R/W	Higher 6 bits (of 14-bit)	0x00	0xFE
0x09	PWMHIGH2L	7-0	R/W	Lower 8 bits (of 14-bit)	0x00	0xFE

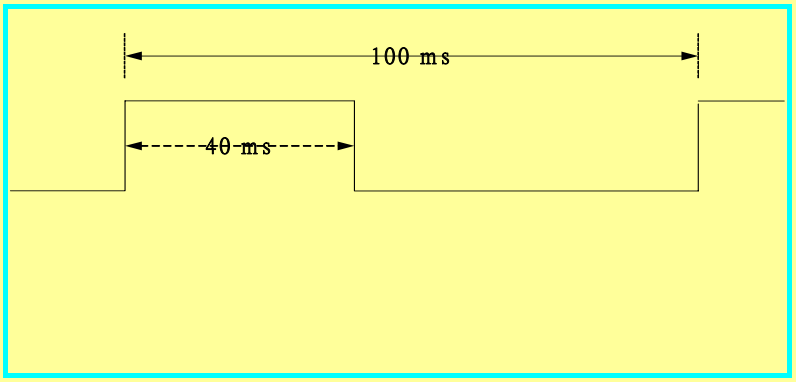
PWM2 Cycle Length (14-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x0A	PWMCYC2H	5-0	R/W	Higher 6 bits (of 14-bit)	0x00	0xFE
0x0B	PWMCYC2L	7-0	R/W	Lower 8 bits (of 14-bit)	0x00	0xFE

PWM3 High Period Length (14-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x0C	PWMHIGH3H	5-0	R/W	Higher 6 bits (of 14-bit)	0x00	0xFE
0x0D	PWMHIGH3L	7-0	R/W	Lower 8 bits (of 14-bit)	0x00	0xFE

PWM3 Cycle Length (14-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x0E	PWMCYC3H	5-0	R/W	Higher 6 bits (of 14-bit)	0x00	0xFE
0x0F	PWMCYC3L	7-0	R/W	Lower 8 bits (of 14-bit)	0x00	0xFE

4.6.3 PWM Programming Sample

In this section gives some programming sample to control PWM module. Please note, ENE does not guarantee these codes in every field application. The following table describes scenario of PWM filed application.

Example
Programming PWM0 with a period 100ms and high period 40ms, that is, duty cycle is 40%. 
Programming model
6. set related GPIO function selection register. GPIOFS08[7] (0xFC01[7]) = 1b 2. clock selection = 4ms PWMCFG[3:0] (0xFE00[3:0]) = 1101b 3. cycle = 4ms * (24+1) PWMCYCL0 (0xFE02) = 0x18 4. duty cycle = (9+1)/(24+1) = 40% PWMHIGH0 (0xFE01) = 0x09

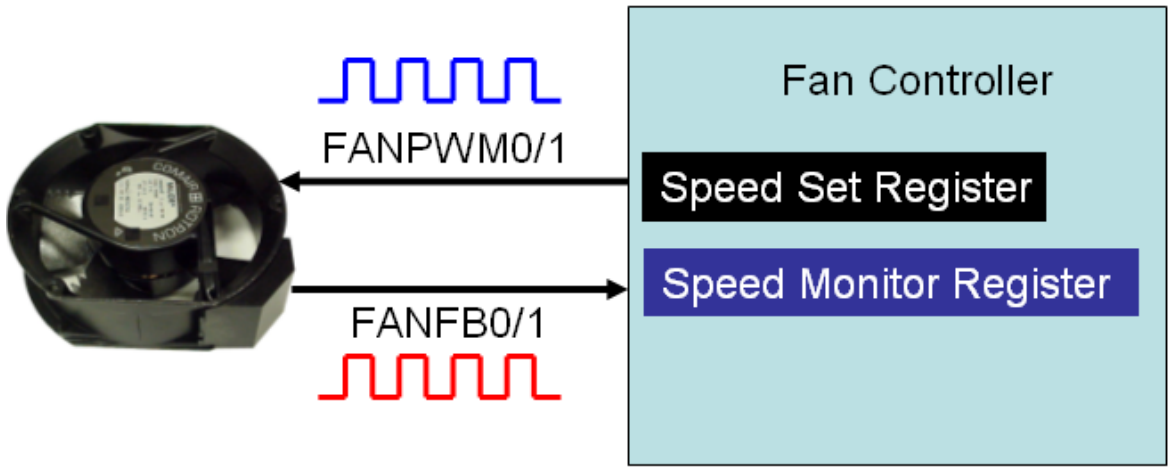
4.7 Fan Controller

4.7.1 Fan Function Description

The KBC provides 2 interfaces with speed monitor for fan control. Two clock selections for fan controller, one is main clock base and the other is fixed 62.5 μs. The fan controller can be configured as a PWM function, as known as FANPWM.

4.7.1.1 Fan Tachometer Monitor

The fan tachometer is implemented by a 12-bit counter with two clock selections, system clock or fixed 62.5 μs. The following figure gives an example for fan speed monitor and control. The KBC uses the pin FANPWM0/1 to drive external fan device, and the fan device feedback the speed via the pin FANFB0/1. The fan controller keeps the speed in the monitor register. The fan controller will compare the speed and check if the current speed is higher or slower than the expected one. If slower, then the controller will increase the frequency to drive FANPWM0/1 automatically, otherwise decrease the frequency. The expected speed can be programmable by F/W.



Here a RPM table is given for programmers. In this table, the information between RPM and value for fan speed set is shown.

RPM	Round/1min	Round/1sec	μs/Round	Value (Set Counter)
6000	6000	100	10000	160 (10000/62.5)
5000	5000	83.33	12000	192 (12000/62.5)
4000	4000	66.667	15000	240 (15000/62.5)
3000	3000	50	20000	320 (20000/62.5)
2000	2000	33.333	30000	480 (30000/62.5)
1000	1000	16.667	60000	960 (60000/62.5)
500	500	8.3	120000	1920(120000/62.5)

RPM (round/min) = 60,000,000 / (FANMON *62.5)

4.7.1.2 FANPWM Function

The fan controller can be used as a 12-bit PWM function. While PWM function applied, the fan controller will refer to the peripheral clock, and the PWM high period and cycle time can be determined as the following formula:

Cycle Length = (PWM cycle register + 1) * peripheral clock

PWM High Period = (PWM high period register + 1) * peripheral clock

Please note, to program the high pulse width of PWM (**FANPWMH0/FANPWML0** and **FANPWMH1/FANPWML1**, i.e., 0xFE26/0xFE27 and 0xFE36/0xFE37), *high-byte first and then low-byte in order.*

4.7.2 Fan Registers Description

Fan0 Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x20	FANCFG0	7	R/W	FAN0 monitor clock selection. **0**: peripheral clock **1**: fixed 62.5 μs (64 μs)	0x00	0xFE
		6	R/W	FAN0 speed monitor counter edge selection. Fan speed monitor counts the high period number between edges. **0**: count between two rising edges. **1**: count between one rising and one falling edge.		
		5	R/W	FANPWM0 cycle width enable **0**: Disable **1**: Enable		
		4	R/W	FANPWM0 enable. **0**: Disable **1**: Enable		
		3	R/W	FAN0 speed monitor interrupt enable **0**: Disable **1**: Enable		
		2	R/W	FAN0 speed monitor timeout error interrupt enable **0**: Disable **1**: Enable		
		1	R/W	Automatic FANPWM control enable. FANCFG0[0] and FANCFG0[4] should be set at the same time to make it work. Please note, once enabled, if fan speed monitor counter smaller than 128, i.e., feedback frequency smaller than 3ms, the automatic fan control will take it as noise. **0**: Disable **1**: Enable		
		0	R/W	FAN0 tachometer monitor enable. **0**: Disable **1**: Enable		

Fan0 Control and Status Register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x21	FANSTS0	7	R/W	Fan auto-load FANCPWM function enable. (FAN0) **0**: Disable **1**: Enable	0x00	0xFE
		6-5	RSV	Reserved		
		4	R/W	FAN0 digital noise filter enable. **0**: Disable **1**: Enable		
		3-2	RSV	Reserved		
		1	R/W1C	Flag of FAN0 speed monitor timeout error **0**: no timeout error **1**: timeout error event		
		0	R/W1C	Flag of FAN0 speed monitor update event. **0**: no update event. **1**: update event		

Fan0 Speed Monitor Counter Value (12-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x22	FANMONH0	3-0	RO	High 4 bits of FAN0 speed monitor counter value	0x0F	0xFE
0x23	FANMONL0	7-0	RO	Low 8 bits of FAN0 speed monitor counter value	0xFF	0xFE

Fan0 Speed Set Counter Value (12-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x24	FANSETH0	3-0	R/W	High 4 bits of target FAN0 speed counter value.	0x00	0xFE
0x25	FANSETL0	7-0	R/W	Low 8 bits of target FAN0 speed counter value.	0x00	0xFE

FANPWM0 High Pulse Width Bits (12-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x26	FANPWMH0	3-0	R/W	High 4 bits of FANPWM0 high pulse width. (FANCFG0[1]=0 only) ***PWM high period =*** ***(PWM high pulse register + 1) * peripheral clock***	0x00	0xFE
0x27	FANPWML0	7-0	R/W	Low 8 bits of FANPWM0 high pulse width. (FANCFG0[1]=0 only) ***PWM high period =*** ***(PWM high pulse register + 1) * peripheral clock***	0x00	0xFE

Current FANPWM0 High Pulse Width Bits (12-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x28	FANCPWMH0	3-0	RO	High 4 bits of current FANPWM0 high pulse width. (FANCFG0[1]=0 only)	0x00	0xFE
0x29	FANCPWML0	7-0	RO	Low 8 bits of current FANPWM0 high pulse width. (FANCFG0[1]=0 only)	0x00	0xFE

FANPWM0 Cycle Length (12-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x2A	FANPWMCH0	3-0	R/W	High 4 bits of Cycle length of FANPWM0 (FANCFG0[5]=1) ***Cycle length = (PWM cycle register + 1) * peripheral clock***	0x00	0xFE
0x2B	FANPWMCHL0	7-0	R/W	Low 8 bits of Cycle length of FANPWM0 (FANCFG0[5]=1) ***Cycle length = (PWM cycle register + 1) * peripheral clock***	0x00	0xFE

FANPWM0 Auto-Load High Pulse Width Bits						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x2C	FANUPWM0	7-4	RSV	Reserved	0x00	0xFE
		3-0	R/W	If auto-load feature enabled (FANSTS0[7]=1), this register value will be auto-loaded into FANCPWMH0 registers and FANCPWML0 will be forced to be zero while monitor timeout occurs		

Fan1 Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x30	FANCFG1	7	R/W	FAN1 monitor clock selection. **0**: peripheral clock **1**: fixed 64 μs	0x00	0xFE
		6	R/W	FAN1 speed monitor counter edge selection. Fan speed monitor counts the high period number between edges. **0**: count between two rising edges. **1**: count between one rising and one falling edge.		
		5	R/W	FANPWM1 cycle width enable **0**: Disable **1**: Enable		
		4	R/W	FANPWM1 enable. **0**: Disable **1**: Enable		
		3	R/W	FAN1 speed monitor interrupt enable **0**: Disable **1**: Enable		
		2	R/W	FAN1 speed monitor timeout error interrupt enable **0**: Disable **1**: Enable		
		1	R/W	Automatic FANPWM control enable. FANCFG1[0] and FANCFG1[4] should be set at the same time to make it work. Please note, once enabled, if fan speed monitor counter smaller than 128, i.e., feedback frequency smaller than 3ms, the automatic fan control will take it as noise. **0**: Disable **1**: Enable		
		0	R/W	FAN1 tachometer monitor enable. **0**: Disable **1**: Enable		

Fan1 Control and Status Register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x31	FANSTS1	7	R/W	Fan auto-load FANCPWM function enable. (FAN1) **0**: Disable **1**: Enable	0x00	0xFE
		6-5	R/W	Reserved		
		4	R/W	FAN1 digital noise filter enable. **0**: Disable **1**: Enable		
		3-2	R/W	Reserved		
		1	R/W	Flag of FAN1 speed monitor timeout error **0**: no timeout error **1**: timeout error event		
		0	R/W	Flag of FAN1 speed monitor update event. **0**: no update event. **1**: update event		

Fan1 Speed Monitor Counter Value (12-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x32	FANMONH1	3-0	R/W	High 4 bits of FAN1 speed monitor counter value	0x0F	0xFE
0x33	FANMONL1	7-0	R/W	Low 8 bits of FAN1 speed monitor counter value	0xFF	0xFE

Fan1 Speed Set Counter Value (12-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x34	FANSETH1	3-0	R/W	High 4 bits of target FAN1 speed counter value.	0x00	0xFE
0x35	FANSETL1	7-0	R/W	Low 8 bits of target FAN1 speed counter value.	0x00	0xFE

FANPWM1 High Pulse Width Bits (12-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x36	FANPWMH1	3-0	R/W	High 4 bits of FANPWM1 high pulse width. (FANCFG1[1]=0 only) ***PWM high period =*** ***(PWM high pulse register + 1) * peripheral clock***	0x00	0xFE
0x37	FANPWML1	7-0	R/W	Low 8 bits of FANPWM1 high pulse width. (FANCFG1[1]=0 only) ***PWM high period =*** ***(PWM high pulse register + 1) * peripheral clock***	0x00	0xFE

Current FANPWM1 High Pulse Width Bits (12-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x38	FANCPWMH1	3-0	RO	High 4 bits of current FANPWM0 high pulse width. (FANCFG1[1]=0 only)	0x00	0xFE
0x39	FANCPWML1	7-0	RO	Low 8 bits of current FANPWM0 high pulse width. (FANCFG1[1]=0 only)	0x00	0xFE

FANPWM1 Cycle Length (12-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x3A	FANPWMCH1	3-0	R/W	High 4 bits of Cycle length of FANPWM1 (FANCFG1[5]=1) ***Cycle length = (PWM cycle register + 1) * peripheral clock***	0x00	0xFE
0x3B	FANPWMCHL1	7-0	R/W	Low 8 bits of Cycle length of FANPWM1 (FANCFG1[5]=1) ***Cycle length = (PWM cycle register + 1) * peripheral clock***	0x00	0xFE

FANPWM1 Update High Pulse Width Bits						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x3C	FANUPWM1	7-4	RSV	Reserved	0x00	0xFE
		3-0	R/W	If auto-load feature enabled (FANSTS1[7]=1), this register value will be auto-loaded into FANCPWMH1 registers and FANCPWML1 will be forced to be zero while monitor timeout occurs		

4.7.3 Fan Programming Sample

In this section gives some programming sample to control FAN module. Please note, ENE does not guarantee these codes in every field application. The following table describes scenario of FAN filed application.

Example
FAN0 @ 4000 rpm with automatic PWM control **FAN1 @ some rpm with fixed PWM control**
Programming model
For FAN0: 6. set related GPIO function select register to enable alternative output. GPIOFS10[2] (0xFC02[2]) = 1b 2. set related GPIO input enable. GPIOIE10[4] (0xFC62[4]) = 1b 3. set FAN0 configuration register FANCFG0 (0xFE20) = 0x93 4. set FAN0 speed monitor counter value FANMONH0 (0xFE24) = 0x00 FANMONL0 (0xFE25) = 0xF0 For FAN1: 6. set related GPIO function select register to enable alternative output. GPIOFS10[3] (0xFC02[3]) = 1b 2. set FAN1 configuration register FANCFG1 (0xFE30) = 0x90 3. set FAN1 speed monitor counter value FANPWMH1 (0xFE36) = 0x03 FANPWML2 (0xFE37) = 0xE8

4.8 General Purpose Timer (GPT)

4.8.1 GPT Function Description

The KBC provides 4 GPTs (General Purpose Timers), two 16-bit timers and two 8-bit timers. These 4 GPTs operate based on 32.768KHz and all timers have the interrupt capability. The GPT is simply a free run counter. While the timer meets the specific value in count register, for instance, 0xFE53 and 0xFE55, an interrupt issues (if interrupt enabled) and the counter reset to be zero.

- GPT0 and GPT1 are 8-bit timers.
- GPT2 and GPT3 are 16-bit timers.

4.8.2 GPT Registers Description

GPT Configuration

Offset	Name	Bit	Type	Description	Default	Bank
0x50	GPTCFG	7-5	RSV	Reserved	0x00	0xFE
		4	R/W	GPT test mode enable. In test mode, the GPT runs with main clock. **0**: Disable **1**: Enable		
		3	R/W	GPT3 counting and interrupt enable. **0**: Disable **1**: Enable		
		2	R/W	GPT2 counting and interrupt enable. **0**: Disable **1**: Enable		
		1	R/W	GPT1 counting and interrupt enable. **0**: Disable **1**: Enable		
		0	R/W	GPT0 counting and interrupt enable. **0**: Disable **1**: Enable		

GPT Pending Flag

Offset	Name	Bit	Type	Description	Default	Bank
0x51	GPTPF	7	WO	Writing “1” to this bit forces GPT3 restart.	0x00	0xFE
		6	WO	Writing “1” to this bit forces GPT2 restart.		
		5	WO	Writing “1” to this bit forces GPT1 restart.		
		4	WO	Writing “1” to this bit forces GPT0 restart.		
		3	R/W1C	Interrupt pending flag of GPT3.		
		2	R/W1C	Interrupt pending flag of GPT2.		
		1	R/W1C	Interrupt pending flag of GPT1.		
		0	R/W1C	Interrupt pending flag of GPT0.		

Reserved						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x52	RSV	7-0	RSV	Reserved	0x00	0xFE

GPT0 Counter Value						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x53	GPT0	7-0	R/W	Once GPT0 counter meets this value, an interrupt issues. GPT0 restart to count from zero.	0x00	0xFE

Reserved						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x54	RSV	7-0	RSV	Reserved	0x00	0xFE

GPT1 Counter Value						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x55	GPT1	7-0	R/W	Once GPT1 counter meets this value, an interrupt issues. GPT1 restart to count from zero.	0x00	0xFE

GPT2 Counter Value (16-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x56	GPT2H	7-0	R/W	High byte of GPT2 counter value. Once GPT2 counter meets this 16-bit value, an interrupt issues. GPT2 restart to count from zero.	0x00	0xFE
0x57	GPT2L	7-0	R/W	Low byte of GPT2 counter value. Once GPT2 counter meets this 16-bit value, an interrupt issues. GPT2 restart to count from zero.	0x00	0xFE

GPT3 Counter Value (16-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x58	GPT3H	7-0	R/W	High byte of GPT3 counter value. Once GPT3 counter meets this 16-bit value, an interrupt issues. GPT3 restart to count from zero.	0x00	0xFE
0x59	GPT3L	7-0	R/W	Low byte of GPT3 counter value. Once GPT2 counter meets this 16-bit value, an interrupt issues. GPT3 restart to count from zero.	0x00	0xFE

4.8.3 GPT Programming Sample

In this section gives some programming sample to control GPT module. Please note, ENE does not guarantee these codes in every field application. The following table describes scenario of GPT filed application.

Example
Programming GPT0 to issue an interrupt every 5ms
Programming model
6. set GPT configuration register, enable GPT0 interrupt. GPTCFG[0] (0xFE50[0]) = 1b 2. fill the GPT counter value. GPT0 (0xFE53) = 0xA6 ; 5000/30 = 0xA6

4.9 Serial Device Interface Controller (SDI)

4.9.1 SDI Function Description

The Serial Peripheral Interface Bus or SPI (often pronounced “spy”) bus is a synchronous serial data link standard designed by Motorola that operates in full duplex mode. Devices communicate in master/slave mode where the master device initiates the data frame. Multiple slave devices are allowed with individual slave select (chip select) lines. The following figure gives an example of how a SPI master works with 3 SPI slaves.

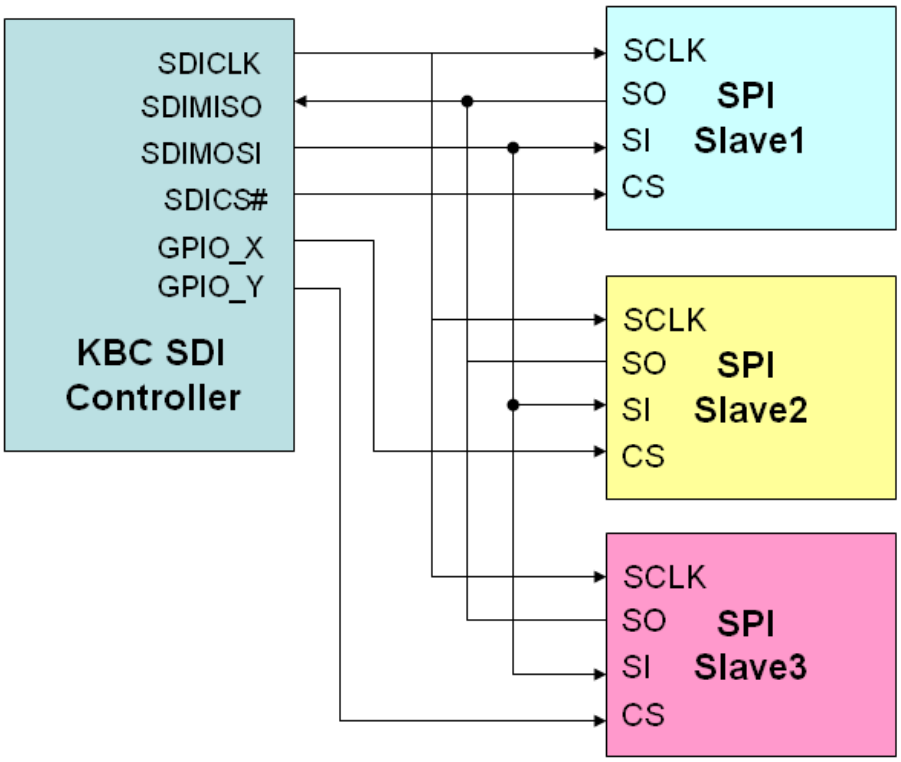
Where

SDICLK — Serial Clock (output from master)

SDIMOSI — Master Output, Slave Input (output from master)

SDIMISO — Master Input, Slave Output (output from slave)

SDICS# — Slave Select (active low; output from master, or said CS, Chip Select)



To support more SPI applications, the KBC introduces a SPI master interface, called **SDI** (Serial Device Interface). With simple programming interface, the F/W can easily communicate with SPI slave devices.

4.9.2 SDI Registers Description

SDI Control/Status Register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x70	SDICSR	7	RO	SDI Idle flag. If this bit set, the SDI is in an idle state. **0**: busy **1**: idle	0x00	0xFE
		6-4	RSV	Reserved		
		3-2	R/W	SDICCLK divider. ***SDICLK frequency = peripheral clock / [(divider +1)*2]***		
		1	R/W	SDIDO/SDIDI Timing. **0**: SDIDO changes data at rising edge of SDICLK. (device latches at falling edge of SDICLK) SDIDI latch data at rising edge of SDICLK. (device changes at falling edge of SDICLK). **1**: SDIDO changes data at falling edge of SDICLK. (device latches at rising edge of SDICLK) SDIDI latch data at falling edge of SDICLK. (device changes at rising edge of SDICLK).		
		0	R/W	SDICS# / SDI data port enable. **0**: Disable **1**: Enable		

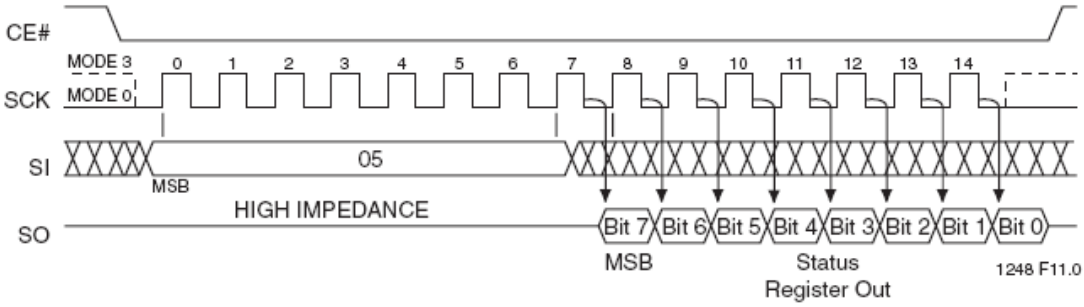
SDI Data Byte Output Port						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x71	SDIBO	7-0	R/W	While SDICSR[7]=0 (SDI not busy), writing to this register forces data output to SDIDO in continuously serial 8 bits. MSB first.	0x00	0xFE

SDI Data Byte Input Port						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x72	SDIBI	7-0	RO	SDIDI reading port.	0x00	0xFE

4.9.3 SDI Programming Sample

In this section gives some programming sample to control SDI module. Please note, ENE does not guarantee these codes in every field application. The following table describes scenario of SDI filed application.

Example
A SPI device is attached. Here is a READ STATUS command.



Programming model

```
GPXAFS00[2:0]   (0xFC0C[2:0]) = 111b;    //Select SDI function pins
GPXDIE00[0] (0xFC6C[0])   = 1b;           //Enable SDI data input
SDICSR (0xFE70) = 0x01;                   //Set CE# low, SPI clock = Peripheral //clock/2
SDIBO   (0xFE71) = 0x05;                  //Transfer CMD(0x05) to device
Wait SDICSR[7]   (0xFE70[7]) = 1b;        //Wait bus idle
SDIBO (0xFE71) = 0x00;       //Transfer dummy byte to device and //device sends status byte to SDI
Wait SDICSR[7] (0xFE70[7])   = 1b;        //Wait bus idle
SDICSR (0xFE70) = 0x00;                   //Set CE# high
Read SDIBI (0xFE72);                      //Get device status
```

4.10 Watchdog Timer (WDT)

4.10.1 WDT Function Description

A Watchdog Timer (WDT) is a hardware timing device that triggers a system reset while the system encounters any unrecoverable situation. The WDT utilizes 32.768KHz for operation. The WDT triggers the system reset in two ways.

- Reset the 8051 microprocessor only.
- Reset the whole logic, except GPIO modules.

Here gives the highlight of WDT features:

- 20 bit Watchdog
- Watchdog password protection.
- Interrupt support.
- WDT LED blinking support.
- New 24 bit timer (TMR) support.

4.10.2 WDT Registers Description

WDT Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x80	WDTCFG	7	R/W	WDT clock source selection **0**: internal 32.768KHz **1**: external 32.768KHz OSC.	0x00	0xFE
		6-3	R/W	WDT disable password Writing 1001**b** to this field will force WDT disable		
		2	R/W	WDT test mode enable. **0**: normal mode **1**: test mode, clock driven by internal 32MHz. (WDTCFG[7] ignore)		
		1	R/W	WDT interrupt enable. **0**: Disable **1**: Enable		
		0	R/W	WDT reset enable. Once WDT resets, two WDT pending flags are clear. **0**: Disable **1**: Enable		

WDT Pending Flag						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x81	WDTPF	7-5	RSV	Reserved	0x00	0xFE
		1	R/W1C	WDT interrupt flag Once the timer counts to half of WDT (0xFE82), an interrupt occurs. If the timer counts to WDT(0xFE82), a WDT reset occurs. **0**: no event **1**: event occurs		
		0	R/W1C	WDT reset flag Once the timer counts to WDT (0xFE82), a WDT reset occurs and this flag is set. **0**: no event **1**: event occurs		

WDT High 8-bit Counter Value (for WDT reset system)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x82	WDT	7-0	R/W	The high 8-bits of WDT counter value. The WDT timer unit is 128ms. Please note, fill this value at least greater than or equal **3** (>=3) for hardware limitation.	0x00	0xFE

WDT Blinking LED Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x83	LEDCFG	7-3	RSV	Reserved	0x00	0xFE
		2-0	R/W	LED Blinking configuration. **0**: LED output keeps high **1**: LED output keeps low 500ms for every 1 sec. **2**: LED output keeps low 500ms for every 2 sec **3**: LED output keeps low 500ms for every 4 sec **4**: LED output keeps low 500ms for every 8 sec		

WDT TMR (24-bit Timer) Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x84	TMR_CFG	7	R/W	TMR enable **0**: Disbale/reset TMR **1**:Enable TMR	0x00	0xFE
		6:3	RSV	Reserved		
		2	RO	TMR interrupt pending flag overflow. While TMR interrupt flag (TMR_CFG[1]) is set and an interrupt event occurs again. This bit will be set and can be clear via writing TMR_CFG[7] with “0”. **0**: no event **1**: event occurs		
		1	R/W1C	TMR interrupt flag. When TMR counter[23:16] is equal to TMR_MATCH register. This bit will be set. **0**: no event **1**: event occurs		
		0	R/W	TMR counter start control. **0**: stop counting **1**: start counting		

WDT TMR (24-bit Timer) Counter Match Value						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x85	TMR_MATCH	7-0	R/W		0x00	0xFE

WDT TMR (24-bit Timer) Counter Value 1						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x86	TMR_V1	7-0	RO	Value for TMR counter[23:16]	0x00	0xFE

WDT TMR (24-bit Timer) Counter Value 2						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x87	TMR_V2	7-0	RO	Value for TMR counter[15:8]	0x00	0xFE

4.10.3 WDT Programming Sample

In this section gives some programming sample to control WDT module. Please note, ENE does not guarantee these codes in every field application. The following table describes scenario of WDT filed application.

Example
Set WDT=512ms to reset system, and an interrupt occurs while WDT=256ms (half of WDT)
Programming model
WDT (0xFE82) = 0x04 ; set WDT=512ms WDTCFG (0xFE80) = 0x03 ; enable interrupt and WDT reset

4.11 Low Pin Count Interface (LPC)

4.11.1 LPC Function Description

The Low Pin Count (LPC) is an interface for modern ISA-free system. The KBC connects to the system via LPC interface. The following LPC cycle types are supported.

Type	Address	Data
LPC I/O Read	16-bit	8-bit
LPC I/O Write	16-bit	8-bit
LPC Memory Read	32-bit	8-bit
LPC Memory Write	32-bit	8-bit
FWH Read	28-bit	8-bit
FWH Write	28-bit	8-bit

4.11.2 LPC I/O Decode Range

Item	Port	Comment
Keyboard Controller	60h/64h	
Embedded Controller	62h/66h (default)	Programmable
Legacy I/O	68h/6Ch, 2Eh/2Fh	
EC Index-I/O	FF29h~FF2Bh/FF2Dh~FF2Fh(default)	2 Sets, Programmable.
Debug Port	80h	Only write cycle support interrupt

4.11.3 LPC Memory Decode Range

Memory Address (hex)	Size	Setting (LPCSCFG[3],LPCFWH[7:6])
000C_0000 ~ 000F_FFFF＊ FFFC_0000 ~ FFFF_FFFF	256K (default)	0b,00b
000C_0000 ~ 000F_FFFF＊ FFF8_0000 ~ FFFF_FFFF	512K	0b,01b
000C_0000 ~ 000F_FFFF＊ FFF0_0000 ~ FFFF_FFFF	1M	0b,10b
000C_0000 ~ 000F_FFFF＊ FFE0_0000 ~ FFFF_FFFF	2M	0b,11b
000C_0000 ~ 000F_FFFF＊ FFC0_0000 ~ FFFF_FFFF	4M	1b,00b
＊ LPC module decodes low memory address only in 256K range.		

4.11.4 FWH Memory Decode Range

Memory Address (hex)	Size	Setting (LPCSCFG[3],LPCFWH[7:6])
00C_0000 ~ 00F_FFFF＊ FFC_0000 ~ FFF_FFFF	256K (default)	0b,00b
00C_0000 ~ 00F_FFFF＊ FF8_0000 ~ FFF_FFFF	512K	0b,01b
00C_0000 ~ 00F_FFFF＊ FF0_0000 ~ FFF_FFFF	1M	0b,10b
00C_0000 ~ 00F_FFFF＊ FE0_0000 ~ FFF_FFFF	2M	0b,11b
00C_0000 ~ 00F_FFFF＊ FC0_0000 ~ FFF_FFFF	4M	1b,00b
＊ LPC module decodes low memory address only in 256K range.		

4.11.5 Index-I/O Port

The KBC provides a method to communicate with the host via legacy I/O port. The host can access the XRAM space inside the KBC. The I/O port is called Index-I/O. Two Index-I/Os are supported and programmable. The registers, **LPCIBAH** and **LPCIBAL** (0xFE92 and 0xFE93), are used to specify the desired I/O port base. To enable the 2nd Index-I/O, the **LPCSCFG**[5], (0xFE90[5]) should be set. The index-I/O base address will be **8** bytes align if the **LPCSCFG**[5] set, otherwise **4** bytes alignment . For example, while the base address is 0xFF2C and LPCSCFG[5] set, the 1st index-I/O address will be 0xFF29 (io_base +1).

The following table collects the port definition for the *host*. The base address of Index-I/O is assumed to be **io_base**.

1st Index-I/O		2nd Index-I/O (LPCSCFG[5]=1)	
XRAM address (high)	io_base+1	XRAM address (high)	io_base+5
XRAM address (low)	io_base+2	XRAM address (low)	io_base+6
XRAM data (high)	io_base+3	XRAM data (high)	io_base+7

Here is an example how to use an Index-I/O.

EC F/W	Host software
1. EC F/W setups the base address, for instance, 0x380. That is, LPCIBAH=0x03 and LPCIBAL=0x80. 2. If the 2nd Index-I/O is needed, turn on the enable bit. That is, LPCSCFG[5]=1 (0xFE90[5]=1).	1. Host setups the desired XRAM address: Port 0x381 = high byte of XRAM address Port 0x382 = low byte of XRAM address 2. And then the host can access the content/data via Port 0x383. 3. If the 2nd Index-I/O required. Port 0x385 = high byte of XRAM address Port 0x386 = low byte of XRAM address Port 0x387 = content/data of XRAM address

4.11.6 Extended I/O Port (Debug Port, Port80)

Developers may use legacy I/O port, 0x80 for debug. The KBC provides a debug interface for this application, called extended I/O port (debug port). The port address can be programmable in the KBC. The host software can use this interface not only for debug but also for special communication with the EC F/W. This interface provides interrupt capability as well. That is, while host accesses this I/O port, an interrupt to 8051 occurs. There is one thing should be reminded. The interrupt feature is only for **I/O-write** to this port, not for I/O-read. Please note, the interrupt capability is controlled in the register **ECCFG**[2] (0xFF04[2]).

4.11.7 LPC Registers Description

LPC SIRQ Configuration for Quiet Mode						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x90	LPCSCFG	7-6	RSV	Reserved	0x00	0xFE
		5	R/W	Enable 2nd index-I/O mode		
		4	R/W	Switch of CIR/User IRQ Switch between CIR and User defined SIRQ, and the SIRQ channel is defined in LPCTCFG[3:0] **0**: User defined SIRQ **1**: CIR SIRQ		
		3	R/W	Memory size 4MB enable (LPC/FWH). If this bit enable, please make sure LPCFWH[7:6]=00**b** **0**: Disable **1**: Enable		
		2	R/W	LPC I/O 2Eh/2Fh decode enable. If enabled, 0xFE9A/0xFE9B are configured to take in charge of LPC I/O 2Eh/2Fh. **0**: Disable **1**: Enable		
		1	Ro	LPC SIRQ mode **0**: Continuous mode **1**: Quiet mode		
		0	WO	Force LPC SIRQ cycle start. Writing "**1**" to this bit forces SIRQ signal low for a pulse.		

LPC SIRQ Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x91	LPCSIRQ	7	R/W	Ignore A22 of FWH cycle. **0**: Disable **1**: Enable	0x00	0xFE
		6	R/W	SCI SIRQ enable **0**: Disable **1**: Enable		
		5	R/W	IRQ12 SIRQ enable **0**: Disable **1**: Enable		
		4	R/W	IRQ1 SIRQ enable **0**: Disable **1**: Enable		
		3-0	R/W	SCI SIRQ channel. **0x00**: no SIRQ **0x01**: IRQ1 **0x02**: SMI# **0x03**: IRQ3 **0x04**: IRQ4 …. **0x0F**: IRQ15		
LPC Index-I/O Base Address (16-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x92	LPCIBAH	7-0	R/W	High byte of LPC index-I/O address	0xFF	0xFE
0x93	LPCIBAL	7-0	R/W	Low byte of LPC index-I/O address (8-byte alignment required)	0x2C	0xFE

LPC Firmware Hub Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x94	LPCFWH	7-6	R/W	Memory size selection (LPC/FWH) **00b**: 256KB **01b**: 512KB **10b**: 1MB **11b**: 2MB	0x00	0xFE
		5	R/W	FWH memory cycle enable **0**: Disable **1**: Enable		
		4	R/W	FWH IDSEL check enable **0**: Disable **1**: Enable		
		3-0	R/W	FWH ID		

LPC Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x95	LPCCFG	7	R/W	LPC memory write protection (including FWH) **0**: Disable **1**: Enable	0x80	0xFE
		6	R/W	Index-I/O port enable **0**: Disable **1**: Enable		
		5	R/W	KBC 60h/64h I/O port enable **0**: Disable **1**: Enable		
		4	R/W	Debug port (port 80) enable **0**: Disable **1**: Enable		
		3	R/W	EC I/O port enable (default port 62h/66h) **0**: Disable **1**: Enable		
		2	R/W	LPC memory cycle enable (not including FWH) **0**: Disable **1**: Enable		
		1	R/W	SIRQ always in continuous mode enable **0**: Disable **1**: Enable		
		0	R/W	LPC CLKRUN# enable **0**: Disable **1**: Enable		

LPC Extended (Debug) I/O Base Address (16-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x96	LPCXBAH	7-0	R/W	High byte of Extended I/O (debug port)	0x00	0xFE
0x97	LPCXBAL	7-0	R/W	Low byte of Extended I/O (debug port)	0x80	0xFE

LPC EC I/O Base Address (16-bit)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x98	LPCEBAH	7-0	R/W	High byte of EC I/O	0x00	0xFE
0x99	LPCEBAL	7-0	R/W	Low byte of EC I/O	0x62	0xFE

LPC I/O 0x2E/0x2F Configuration and Status (LPCSCFG[2]=1)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x9A	LPC2ECFG	7-4	RSV	Reserved	0x00	0xFE
		3	RO	The previous access type of 2Eh/2Fh **0**: Read **1**: Write		
		2	R/W1C	Interrupt flag of accessing 2Fh I/O. **0**: no event **1**: event occurs		
		1	R/W	2Fh I/O interrupt enable If this bit set, while host accesses 2Fh I/O, an interrupt will issue. **0**: Disable **1**: Enable		
		0	R/W	Decode 2Eh/2Fh I/O enable. **0**: Disable **1**: Enable		

LPC User Defined SIRQ Configuration (LPCSCFG[2]=0)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x9B	LPCTCFG	7-6	RSV	Reserved	0x00	0xFE
		5	R/W	User defined SIRQ Setting. **0**: Low **1**: High		
		4	R/W	User defined SIRQ channel enable **0**: Disable **1**: Enable		
		3-0	R/W	User defined SIRQ channel number **0x00**: no SIRQ **0x01**: IRQ1 **0x02**: SMI# **0x03**: IRQ3 **0x04**: IRQ4 …. **0x0F**: IRQ15		

LPC I/O 0x2E Read Port Register (LPCSCFG[2]=1)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x9B	LPCTCFG	7-0	RO	Host writes data to I/O port 0x2E, EC F/W could read data from this register.	0x00	0xFE

LPC Read/Write Data of I/O 0x2F (LPCSCFG[2]=1)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x9C 0x9C	LPC2FDAT	7-0	R	Host writes data to I/O port 0x2F, EC F/W could read data from this register.	0x00	0xFE
		7-0	W	If host issue any read access to I/O port 0x2F, the host will get the data which kept in this register		

LPC I/O 0x68/0x6C Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x9D	LPC68CFG	7	R/W	LPC decode I/O port 68h/6Ch enable **0**: Disable **1**: Enable	0x00	0xFE
		6-2	RSV	Reserved		
		1	R/W	IBF interrupt enable Interrupt issues while IBF rising (LPC write I/O 68h/6Ch) **0**: Disable **1**: Enable		
		0	R/W	OBF interrupt enable Interrupt issues while OBF falling (LPC read I/O 68h) **0**: Disable **1**: Enable		

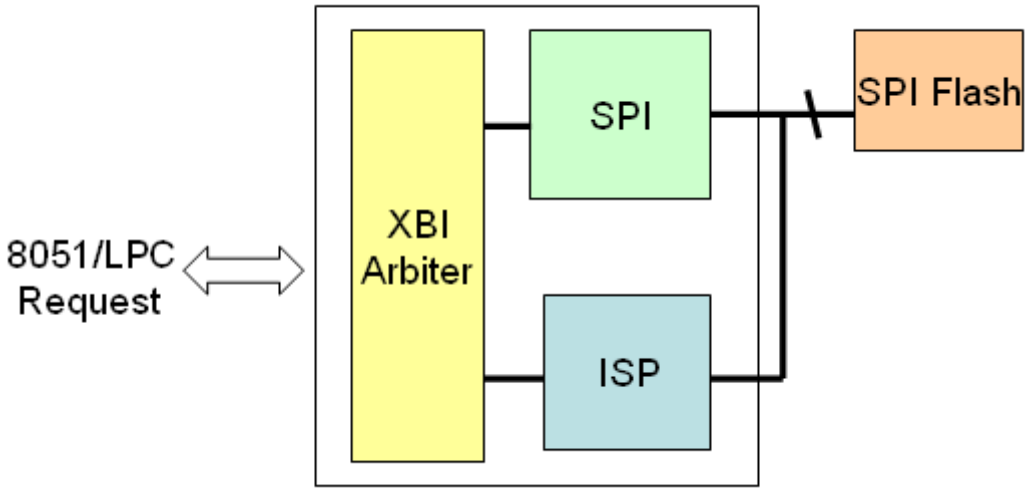
LPC I/O 0x68/0x6C Configuration and Status Register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x9E	LPC68CSR	7	R/W1C	I/O 68h/6Ch busy flag. EC F/W can write "**1**" to clear this flag. A write cycle to port 6Ch with data 0xFF also clear this flag **0**: not busy **1**: busy	0x00	0xFE
		6	RO	Indicator of write port. **0**: write 68h occurs **1**: write 6Ch occurs.		
		5-4	RSV	Reserved		
		3	R/W1C	IBF interrupt flag Interrupt flag while IBF rising (LPC write I/O 68h/6Ch) **0**: no event **1**: event occurs		
		2	R/W1C	OBF interrupt flag Interrupt flag while OBF falling (LPC read I/O 68h) **0**: no event **1**: event occurs		
		1	R/W1C	IBF of port 68h/6Ch		
		0	R/W1C	OBF of port 68h/6Ch		

LPC I/O 0x68/0x6C Data Register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x9F 0x9F	LPC68DAT	7-0	R	Host writes data to I/O port 68h/6Ch, EC F/W could read data from this register.	0x00	0xFE
		7-0	W	If host issue any read access to I/O port 68h/6Ch, the host will get the data which kept in this register		

4.12 X-Bus Interface (XBI)

4.12.1 XBI Function Description

The KBC implements a XBI module to handle the related request from 8051/LPC to flash device. The following figure gives the illustration.



The XBI module also takes the responsibility for the In-System-Programming (ISP) mechanism to update system BIOS. The detail steps to update system BIOS via ISP mode, please refer to the section of ISP. Here gives the feature of XBI module.

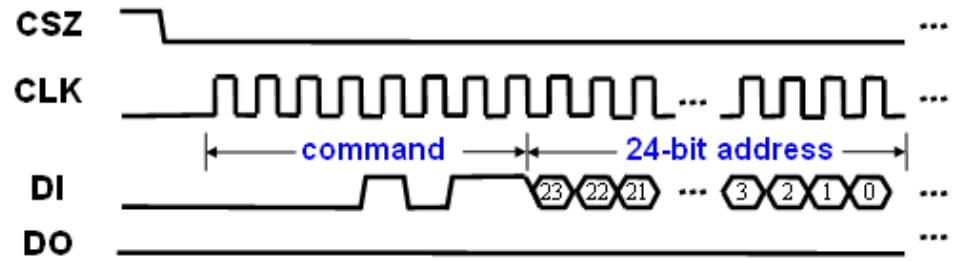
- Two 8051 code segments, one for 16K and the other for 48K.
- XBI arbiter to handle the transaction of 8051 and LPC request.
- XBI pre-fetch code mechanism support for better performance.
- Flash write-protection support.
- ISP flash update support.

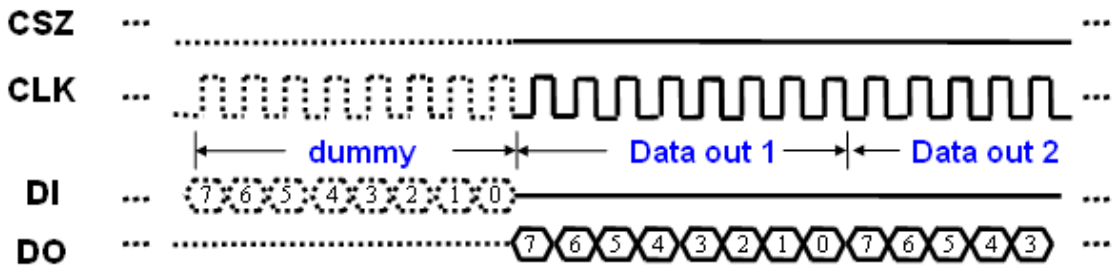
4.12.2 XBI SPI Enhancement

The 8051 microprocessor executes machine codes from SPI flash and the performance is determined by the read operation. To enhance the performance of SPI flash fetching, 3 special read protocols are introduced. They are **Offset_Read**, **Short_Read** and **Dual_Input** protocols. The following sections give a brief introduction.

4.12.2.1 Original Read Protocol

An original SPI read protocol to read flash is as the following figure. A chip select is asserted to the specific flash device and the SPI flash controller drives clock out. A 24-bit address phase follows 8-bit command phase. After a dummy phase, the SPI flash device returns data. Please note, the KBC currently supports 2MB (20bit) size SPI flash, therefore, address bit, A23~A21 are all zero.

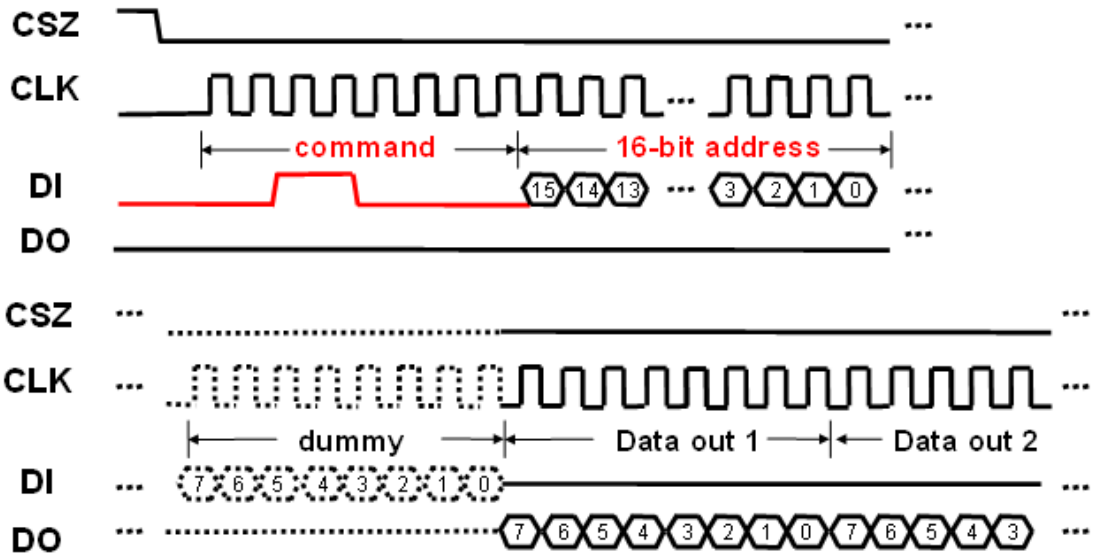




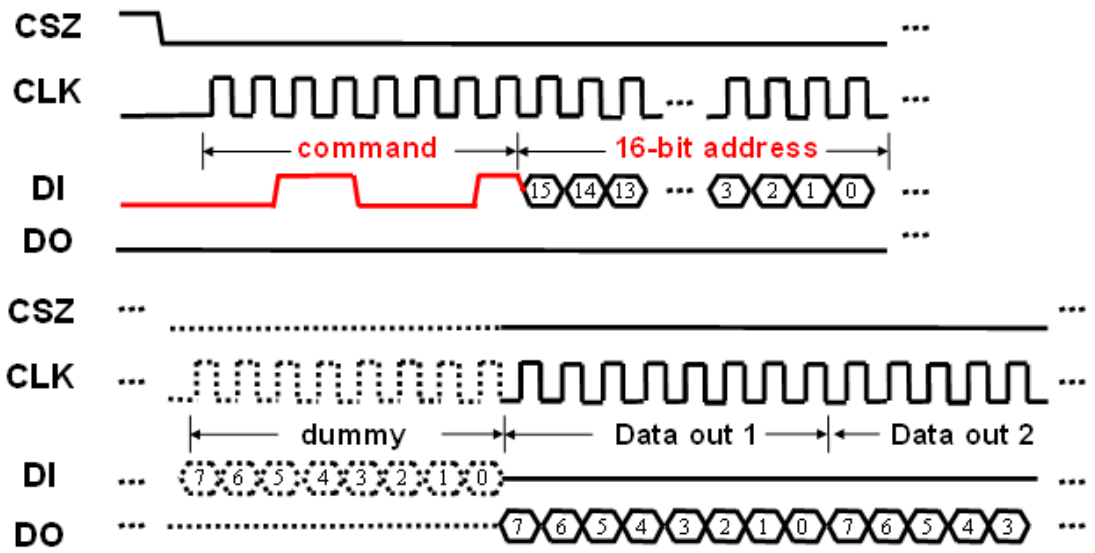
4.12.2.2 Short Read Protocol

The 8051 microprocessor supports 16-bit address (64KB) internally. If the code fetching is within 64KB or 128KB, a **Short_Read** command will enhance the performance by reducing address phase to be 16-bit. The protocol is transparent to EC F/W. The F/W just takes care of the segment switch by **XBISEG0** and **XBISEG1** (0xFEA0 and 0xFEA1). The high nibble of command Short_Read is programmable. Two command bytes are defined, **0x30** and **0x31**, by default, that is, the high nibble is 0x3. The command 0x30 is used while the 1st 64KB code segment is selected and 0x31 is for the 2nd 64KB code segment. Please note, the Short_Read command is not a generic SPI flash command and SPI flash device should support it as well.

Command byte=0x30, A23~A17=0 and A16=0. (1st 64KB code segment selected)

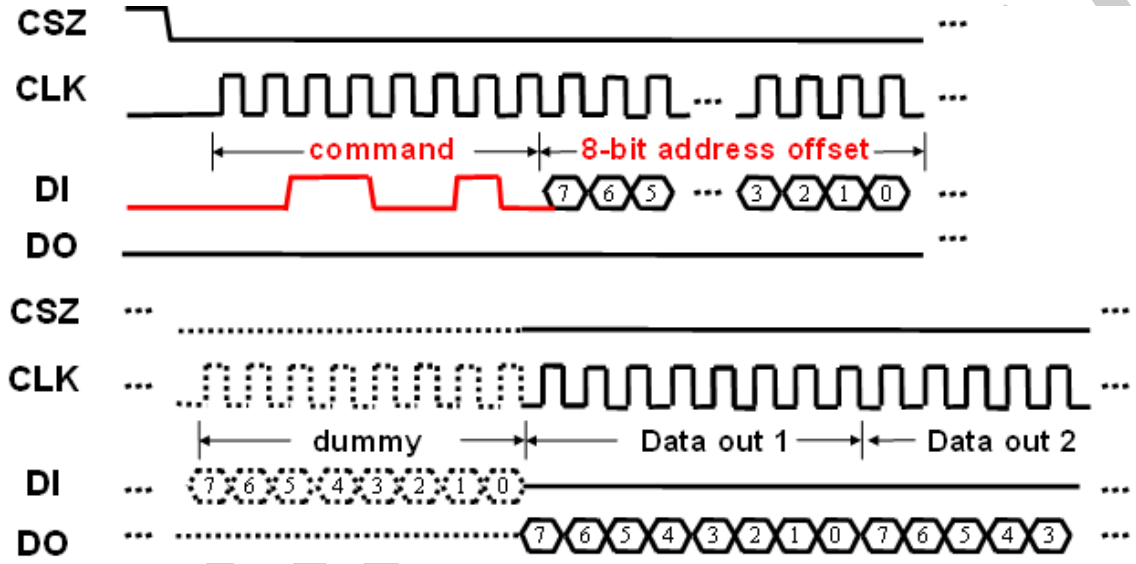


Command byte=0x31, A23~A17=0 and A16=1. (2nd 64KB code segment selected)



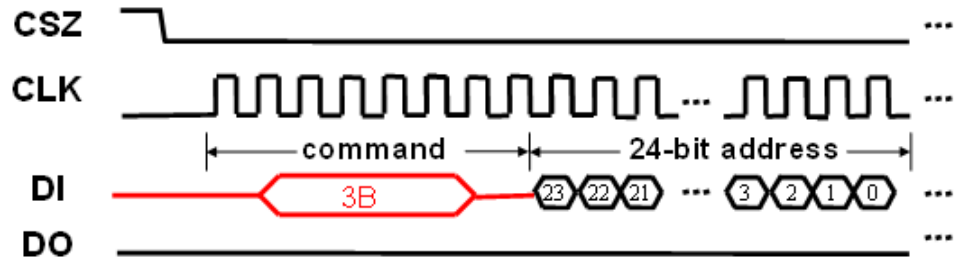
4.12.2.3 Offset Read Protocol

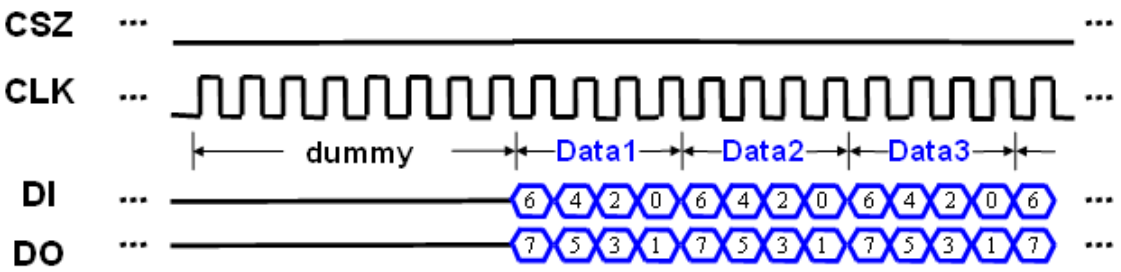
To shorten the address phase and to improve the SPI flash performance, another new protocol is introduced. If branch instruction used often, an **Offset_Read** protocol will improve the performance while fetching code. Once the protocol is used, the address phase will be 8-bit long. If the branch is over 128-byte long, the Offset_Read will not be utilized. Just like Short_Read command, the Offset_Read is also transparent to EC F/W. By default, the command byte is 0x32 and the high nibble of this command is programmable. Please note, the Offset_Read command is not a generic SPI flash command and SPI flash device should support it as well.



4.12.2.4 Dual Input Protocol

Besides the two new protocols mentioned before, an improvement of data phase to increase performance is introduced. The method is called Dual Input mode. In this mode, data output pin works as another input. The bit stream is shown as the following figure. Please note, the high nibble of this protocol is fixed to be 0x3. This is not a standard protocol and SPI flash devices should implement this feature to make it work.





4.12.3 XBI Registers Description

8051 Address Segment 0 Mapping Configuration (0x0000~0x3FFF)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA0	XBISEG0	7	R/W	8051 code segment SEG0 remapping enable. **0**: Disable **1**: Enable	0x00	0xFE
		6	RSV	Reserved		
		5-0	R/W	SEG0 XBI Address SEG0 XBI Address = XBISEG0[5:0]*16K + 8051 Address[13:0]		

8051 Address Segment 1 Mapping Configuration (0x4000~0xFFFF)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA1	XBISEG1	7	R/W	8051 code segment SEG1 remapping enable. **0**: Disable **1**: Enable	0x00	0xFE
		6-4	RSV	Reserved		
		3-0	R/W	SEG1 XBI Address SEG1 XBI Address = XBISEG1[3:0]*64K + 8051 Address[15:0]		
Reserved						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA2	RSV	7-0	RSV	Reserved	0x00	0xFE

LPC Read Buffer Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA3	XBI_LPBCFG	7	R/W	LPC buffer read enable **0**: disable (default) **1**: enable	0x00	0xFE
		6	R/W	LPC buffer auto pre-fetch next 16-byte **0**: disable (default) **1**: enable		
		5-0	RSV	Reserved		

XBI XIO Enable						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA4	RSV	7-0	RSV	Reserved	0x00	0xFE

XBI Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA5	XBICFG	7	RSV	Reserved	0x00	0xFE
		6	R/W	8051 instruction fetch (sustaining access) **0**: Disable **1**: Enable		
		5-0	RSV	Reserved		

XBI E51CS# Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA6	XBICS	7-6	RSV	Reserved	0x00	0xFE
		5	R/W	XBI arbitration priority. **0**: Disable **1**: Enable		
		4	R/W	Reset code segment enable. Once the 8051 reset, the code segment SEG0 or SEG1 can be reset if the corresponding code segment enabled. (XBISEG0[7]/XBISEG1[7]) **0**: Disable **1**: Enable		
		3	R/W	2K-XRAM as code memory enable. To enhance the code fetching, users can use 2K-XRAM as code memory. Please note, users should move codes from flash to XRAM, jump to XRAM and then enable this bit. **0**: Disable **1**: Enable		
		2	R/W	Reset XBI arbiter while in idle/stop mode. **0**: Disable **1**: Enable		
		1	R/W	EHB fast accessing enable. Enable this bit gets better performance in EHB. **0**: Disable **1**: Enable		
		0	RSV	Reserved		

XBI Write Enable						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA7	XBIWE	7-0	R/W	XBI write command. **00h**: exit SRAM test mode **C5h**: enter SRAM test mode	0x00	0xFE

XBI SPI Flash Address (22-bit) = [SPIA2(6bit) : SPIA1(8bit) : SPIA0(8bit)]						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA8	SPIA0	7-0	R/W	SPI Address lowest 8-bits (A7:A0)	0x00	0xFE
0xA9	SPIA1	7-0	R/W	SPI Address middle 8-bits (A15:A8)	0x00	0xFE
0xAA	SPIA2	5-0	R/W	SPI Address upper 6-bits (A21:A16)	0x00	0xFE

XBI SPI Flash Output/Input Data Port						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xAB	SPIDAT	7-0	R/W	Input (read) / Output (write) data port of SPI flash interface.	0x00	0xFE

XBI SPI Flash Command Port						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xAC	SPICMD	7-0	R/W	Commands support for SPI flash. Writing this register will force the SPI protocol start. Please note, the address phases must be prior to command phase. **SPI command support:** 01h: Write Status Register 02h: Byte Program 03h: Read 04h: Write Disable 05h: Read Status Register 06h: Write Enable 0Bh: High Speed Read 20h: Sector Erase (SST) 3Bh: Fast Read Dual Output (Windbond, AMIC) 50h: Enable Write Status Register (SST) 52h: Block Erase (SST) 60h: Chip Erase (SST) B9h: Power Down C7h: Chip Erase (PMC, NexFlash) D7h: Sector Erase (PMC) D8h: Block Erase (PMC, NexFlash)	0x00	0xFE

XBI SPI Flash Configuration/Status Register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xAD	SPICFG	7	R/W	Fast read dual output mode enable. Please note, if this bit set, the SPICFG[2] will be ignored. **0**: Disable **1**: Enable	0x00	0xFE
		6	R/W	SPI flash offset read command enable. (32h) **0**: Disable **1**: Enable		
		5	R/W	SPI flash short read command enable. (31h, 30h) **0**: Disable **1**: Enable		
		4	R/W	SPICS# force low **0**: SPICS# high **1**: SPICS# low		
		3	R/W	Write enable of **SPICMD** register,0xFEAC. **0**: Disable **1**: Enable		
		2	R/W	Dummy byte of read command. **0**: Disable **1**: Enable		
		1	RO	SPI busy flag. **0**: not busy **1**: busy		
		0	R/W	Automatic SPI status check after a SPICMD issued, until SPI busy flag, clear, i.e., (SPICFG[1]=0). **0**: Disable **1**: Enable		

XBI SPI Flash Output Data Read Back Port						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xAE	SPIDATR	7-0	R/W	Reflection of **SPIDAT** ,0xFEAB register.	0x00	0xFE

XBI SPI Flash Configuration 2						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xAF	SPICFG2	7-4	RSV	Reserved	0x03	0xFE
		3-0	R/W	High nibble of Offset Command / Short Read Command		

4.13 Consumer IR Controller (CIR)

4.13.1 CIR Function Description

The KBC embeds with a native hardware Consumer IR controller, which connects to system via LPC interface. Popular protocols are supported, such as RC-5/RC-6/NEC/RLC. The CIR controller handles the protocol of RC-5/RC-6/NEC/RLC for receiving, and only RLC for transmit. IRQ and I/O port are implemented. An extended function is implemented to support learning application. The basic features are list as the following table.

	925B/926B	926D
RX carrier demodulation	V	V
TX carrier modulation	X	V
RX protocol support	RC5/RC6/NEC/RLC	RC5/RC6/NEC/RLC
TX protocol support	RLC	RLC
RX carrier frequency measurement	X	V

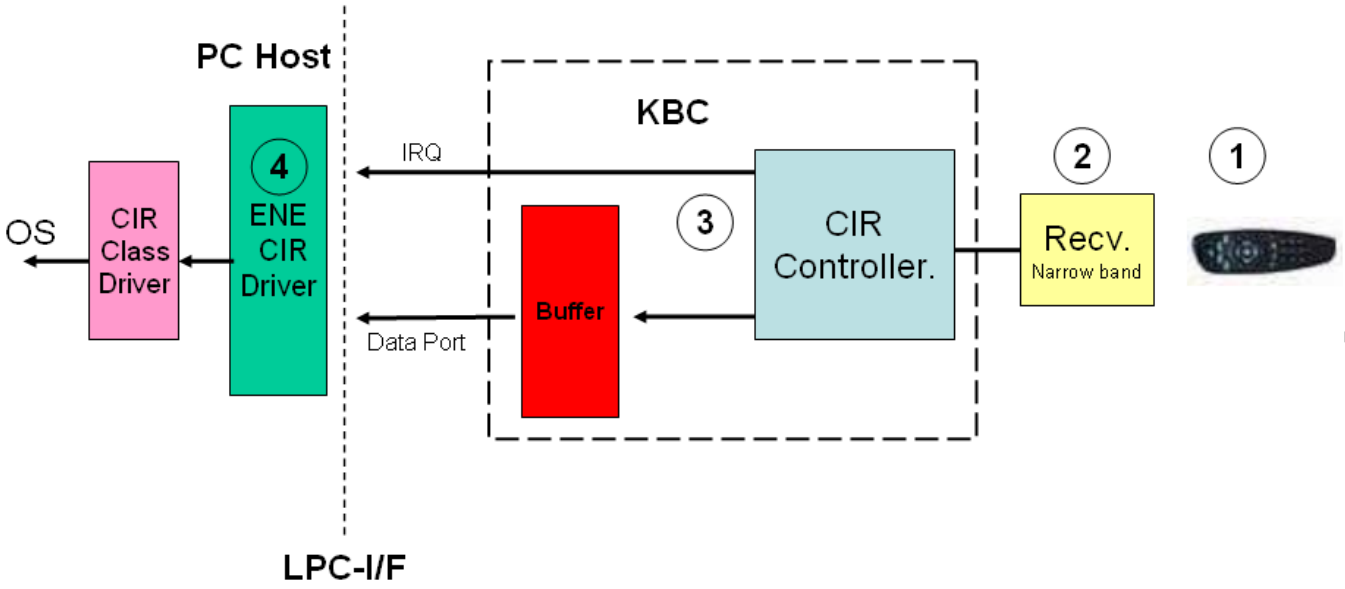
A SIRQ channel can be assigned for CIR application. The related programming registers are summarized as following table.

Register	Description
LPCSCFG[4] (0xFE90[4])	SIRQ selection for LPCTCFG[3:0] (0xFE9B[3:0]) **0**: User defined IRQ **1**: CIR IRQ enable
LPCTCFG[3:0] (0xFE9B[3:0])	SIRQ channel number. **0x00**: IRQ0 **0x01**: IRQ1 … **0x0F**: IRQ15

Here is the features highlight.

- Native hardware protocol decoder, such as RC5/RC6/NEC and RLC.
- I/O and IRQ resource for CIR controller.
- Support **2** sets of RX/TX in one chip, and RX/TX works simultaneously.
- RX carrier demodulation/ TX carrier modulation support.
- Wide range of carrier frequency support, **15K~1MHz**. (The carrier frequency is 30K~60KHz in normal application)
- More flexible in carrier sample frequency, **1 μs.~128 μs** (The sample frequencies are 25, 50 and 100 μs for normal application).
- Remote controller learning support.

The following figure shows an example how a CIR controller works with narrow band receiver.



Here gives the guidance for programming CIR.

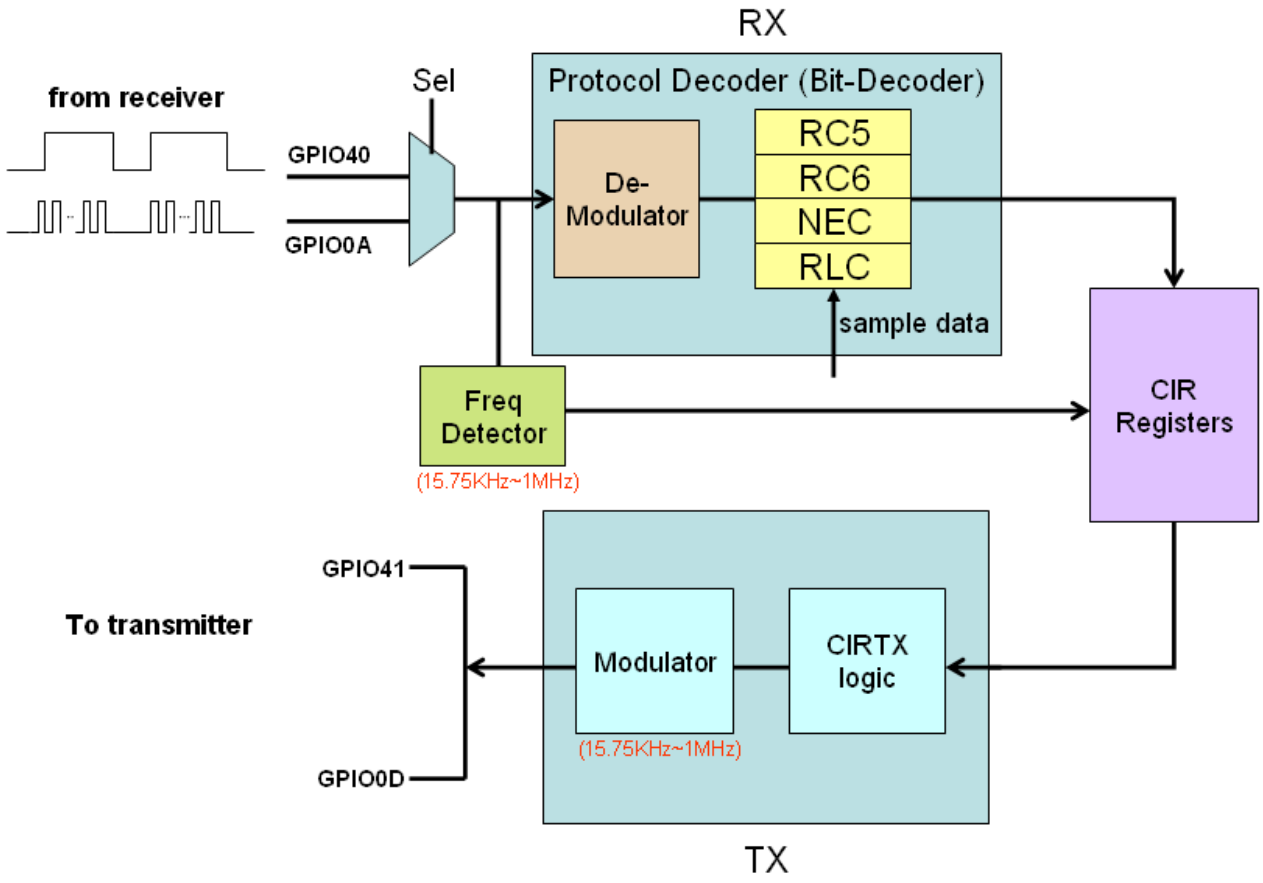
For Receive	For Transmit
1. Select protocol via setting CIRCFG2, i.e., 0xFEC1 2. According to the selected protocol, setup CIRHIGH/CIRBIT/CIRSTART/CIRSTART2, i.e., 0xFEC3~0xFEC6 3. Enable protocol and other configuration setting via CIRCFG, .i.e., 0xFEC0 4. EC F/W waits for data-in by pooling or interrupt.	1. Select RLC protocol and enable via setting CIRCFG, .i.e., 0xFEC0. 2. Writing to CIRRLC_OUT0, 0xFEC9, will start to transmit. 3. If CIRRLC_OUT0 (0xFEC9) and CIRRLC_OUT1 (0xFECA) are written at the same time, it start to transmit CIRRLC_OUT0 and then CIRRLC_OUT1. 4. If only CIRRLC_OUT0 (0xFEC9) is written, the hardware will transmit CIRRLC_OUT0 first and then CIRRLC_OUT1. 5. Each byte transmit completion, an interrupt will fire.

4.13.2 CIR Block Diagram

The CIR controller supports two RX ports (GPIO40/GPIO0A) and two TX ports (GPIO41/GPIO0D). A register bit, **CIRCFG2**[5] (0xFEC1[5]), is used to determine RX source. For example, if CIRCFG2[5]=0, GPIO40 is the RX source, otherwise GPIO0A. The TX port is selected according to the GPIO function selection register. The following table gives an example of RX/TX combination.

	GPIOFS08[5]=0b, GPIOFS[1]=1b	GPIOFS08[5]=1b, GPIOFS[1]=0b
CIRCFG2[5]=0b	(RX,TX)=(GPIO40,GPIO41)	(RX,TX)=(GPIO40,GPIO0D)
CIRCFG2[5]=1b	(RX,TX)=(GPIO0A, GPIO41)	(RX,TX)=(GPIO0A, GPIO0D)

The CIR controller could detect the carrier frequency and demodulate the carrier. This provides a *learning* feature for CIR application. The frequency detection range is from 15.75KHz to 1MHz. After demodulation, the CIR controller handles remote signals with hardware decoder which supports ***RC5/RC6/NEC/RLC*** protocols. If transmit function needed, the CIR controller could modulate the carrier and send it out via GPIO41/GPIO0D. The output carrier frequency range is the same as input, that is, 15.75KHz~1MHz. *The RX and TX can work simultaneously in the current design*. The following diagram gives more detail about CIR controller.



4.13.3 CIR Remote Protocol

In this section, brief introduction of protocols supported in the CIR is given. Four protocols are supported, Philips RC5/RC6, NEC and Run-Length-Code. Only features and protocol definition listed. For more detail please refer to the related specifications.

4.13.3.1 Philips RC5 Protocol

Here highlights the features of Philips RC5 protocol.

- Manufacturer Philips.
- Carrier frequency 36KHz.
- Bi-phase coding.
- 5 bits address / 6 bits command lengths

RC5 Protocol													
Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10	Bit11	Bit12	Bit13	Bit14
S1	S2	T	Address					Command					

S1/S2: start bits, always “1”
T: toggle bit, This bit is inverted every time a key is released and pressed again.
Address: IR device address, MSB first.
Command: IR command, MSB first.

4.13.3.2 Philips RC6 Protocol

Here highlights the features of Philips RC6 protocol.

- Manufacturer Philips.
- Carrier frequency 36KHz.
- Bi-phase coding.
- 5 bits address
- Variable command lengths based on the operation mode.

RC6 Protocol																					
LS	SB	MB2	MB1	MB0	T	A 7	A 6	A 5	A 4	A 3	A 2	A 1	A 0	C 7	C 6	C 5	C 4	C 3	C 2	C 1	C 0	
Header						Control								Information								SF
Header Phase (ENE CIR)						*Data Phase (ENE CIR)*																

LS: Leader symbol
SB: Start bit, always “1”
MB2-MB0: Mode bits, operation mode selection.
T: Trailer bit, this bit can be served as a toggle bit.
A7-A0: Address
C7-C0: Command
SF: Signal free time, 2.666ms.

4.13.3.3 NEC Protocol

Here highlights the features of NEC protocol.

- Manufacturer NEC.
- Carrier frequency 38KHz.
- Pulse distance modulation.
- 8 bit address / 8 bit command length
- Address/Command transmitted twice.
- Total transmit time is constant.

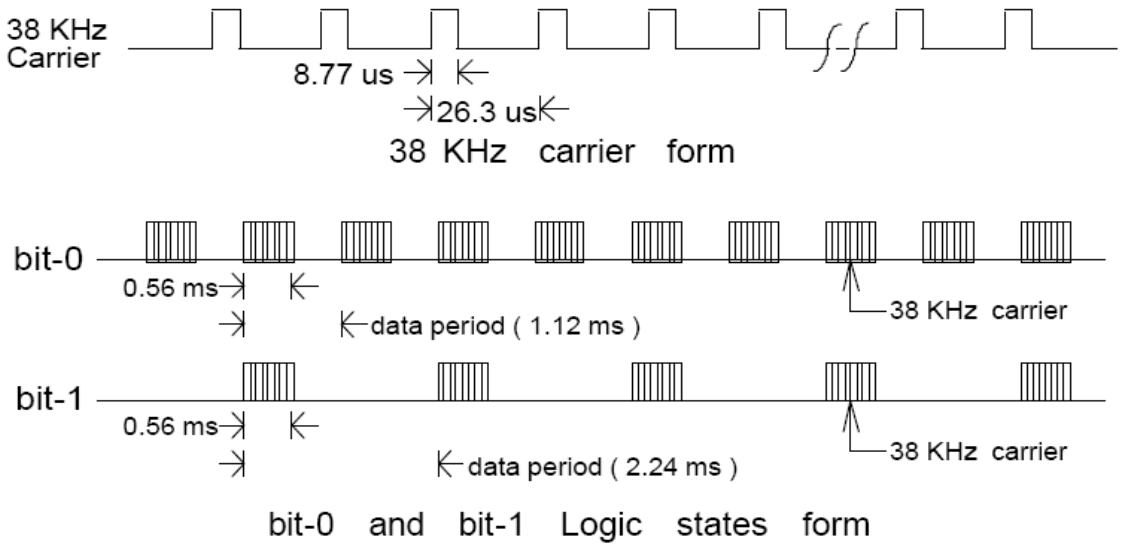
NCE Protocol					
AGC burst	space	Address	～Address	Command	～Command
9ms	4.5ms	8bit	8bit	8bit	8bit

AGC burst: set gain of IR remote controller, 9ms long
Space: follow by AGC burst, 4.5ms.
Address: 8-bit address, LSB first.
~Address: inverted 8-bit address, LSB first.
Command: 8-bit command, LSB first.
~Command: inverted 8-bit command, LSB first

4.13.4 CIR Automatic Carrier Frequency Detection and Modulation

To support learning feature, wide-band transmitter and receiver will be used in a system. The KBC introduces a new mechanism to provide carrier frequency information of wide-band receiver to the host.

The CIR controller needs to be programmed with two parameters for the detection. Register, **CIRCAR_PULS** is used to determine these two parameters. **CIRCAR_PULS**[7:4] keeps the discard number of carrier pulse and **CIRCAR_PULS**[3:0] keeps the average number to detect. The **CIRCAR_PULS**[7:4] tells the controller to discard the specific number of carrier pulse from the beginning. The controller then gets the average number of carriers pulse as sample data and analyzes. The detection of carrier period is kept in **CIRCAR_PRD**[6:0], and the valid flag is kept in **CIRCAR_PRD**[7]. Please note, the detection range is from *15.75KHz~1MHz*. (The general application is from 30K~60KHz).



Here gives an example as the above waveform. Bit stream with 38KHz carrier is shown as bit-0. Each bit is 0.56ms in length and 38KHz carrier period is 26.3 μs, that is, there will be about 21 carrier pulses in a bit. If **CIRCAR_PULS**[7:4]=5 and **CIRCAR_PULS**[3:0]=10, once the detection enabled, the CIR controller will get 6th carrier pulse as the first one and analyze the sequential 10 pluses. The detection result can be obtained via register **CIRCAR_PRD**.

The related registers for automatic carrier frequency detection are listed as following.

Register	Address	Description
CIRCFG2[5:4]	0xFEC1[5:4]	Bit5=1, select wide-band as bit-decoder input. Bit4=1, enable wide-band frequency detection
CIRCAR_PULS	0xFECB	CIRCAR_PULS[7:4] = discard number of carrier pulse CIRCAR_PULS[3:0] = average number of carrier pulse
CIRCAR_PRD	0xFECC	Detection of wide-band carrier period
CIRCAR_HPRD	0xFECD	Detection of wide-band carrier period, pulse width high.

The KBC provides the modulation ability for RLC transmit. The carrier frequency of modulation can be programmable. Before the carrier modulation, the programmer should notice the modulation polarity. That is, if the data bus (TX) is kept low in idle state, only data in high state will be modulated and the bit, **CIRMOD_PRD**[7], should be "**1**".

The related registers for RLC modulation is summarized as below.

Register	Address	Description
CIRCFG[7]	0xFEC0	RLC output modulation enable.
CIRMOD_PRD	0xFECE	CIRMOD_PRD[7] = modulation polarity selection CIRMOD_PRD[6:0] = modulation carrier period
CIRMOD_HPRD	0xFECF	CIRMOD_HPRD[6:0] = modulation carrier period, pulse width high.

4.13.5 CIR Registers Description

CIR Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xC0	CIRCFG	7	R/W	Output carrier modulator for RLC (TX) **0**: Disable **1**: Enable	0x00	0xFE
		6	R/W	Output polarity reversed for RLC. (TX) **0**: Disable **1**: Enable		
		5	R/W	Interrupt while transmit completes with RLC protocol. (TX) **0**: Disable **1**: Enable		
		4	R/W	Output enable for RLC protocol. (TX) Once the data filled into CIRRLC_OUT1 (0xFECA), the controller starts the transmit with RLC protocol **0**: Disable **1**: Enable		
		3	R/W	Input carrier demodulator. (RX) **0**: Disable **1**: Enable		
		2	R/W	Input polarity reversed. (RX) **0**: Disable **1**: Enable		
		1	R/W	Interrupt enable. (RX) Two conditions issue interrupt. a. After decode a byte in RX b. Once receive the “Repeat” in NEC protocol **0**: Disable **1**: Enable		
		0	R/W	Protocol decode enable. (RX) **0**: Disable **1**: Enable, protocol is determined by CIRCFG2[3:0], 0xFEC1.		

CIR Configuration 2						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xC1	CIRCFG2	7	R/W	Fast sample (data phase, not leader phase) enable for input signal. If this bit set, the sample period changes. For RC5/RC6, period changes from 30 μs to 16 μs For NEC, period changes from 64 μs to 30 μs **0**: Disable **1**: Enable	0x00	0xFE
		6	R/W	Fast sample (leader phase) enable for input signal. If this bit set, the sample period changes. For RC6, period changes from 64 μs to 30 μs **0**: Disable **1**: Enable		
		5	R/W	Input selection for protocol decoder (bit-decoder) **0**: from GPIO40 **1**: from GPIO0A		
		4	R/W	Frequency detection enable. **0**: Disable **1**: Enable		
		3-0	R/W	CIR Protocol selection. (valid while CIRCFG[0]=1) **0**: RLC **1**: RC5 **2**: RC6 **3**: NEC **others**: reserved.		

CIR Pending Flag and Status						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xC2	CIRPF	7	RO	Hardware RX idle state. **0**: not idle state **1**: idle state	0x00	0xFE
		6	R/W	Hardware TX (RLC) idle state. **0**: not idle state **1**: idle state		
		5-4	RSV	Reserved		
		3	R/W1C	Pending flag of RLC transmit complete **0**: no event **1**: event occurs		
		2	R/W1C	Pending flag of RLC receive counter overflow **0**: no event **1**: event occurs		
		1	R/W1C	Pending flag of NEC repeat protocol **0**: no event **1**: event occurs		
		0	R/W1C	Pending flag of data-in This bit is set while data received and stored in **CIRDAT_IN**. **0**: no event **1**: event occurs		

Value for High Pulse Width						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xC3	CIRHIGH	5-0	R/W	This register determines the high pulse width of a “logic bit”. High pulse width = Decoder sample period * **CIRHIGH**	0x00	0xFE

Value for Bit Width(RC5/RC6) / Logic Bit-One (NEC)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xC4	CIRBIT	6-0	R/W	This register determines the bit width of a “logic bit”. (RC5/RC6) Bit width = Decoder sample period * **CIRBIT** This register determines the “logic bit-one”. (NEC) Logic bit-one = Decoder sample period * **CIRBIT**	0x00	0xFE

Value for Leader Pulse Width (RC6/NEC) for Normal Packet						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xC5	CIRSTART	6-0	R/W	This register determines the leader pulse width for normal packet (RC6/ENC) Leader pulse width = Decoder sample period * **CIRSTART**	0x00	0xFE

Value for Tailer Bit Width (RC6) / Leader Width of Repeat Packet (NEC)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xC6	CIRSTART2	6-0	R/W	This register determines the bit width of trailer (RC6) trailer bit width = Decoder sample period * **CIRSTART2** This register determines the leader width of repeat packet (NEC) Leader width(repeat) = Decoder sample period * **CIRSTART2**	0x00	0xFE

CIR Decode Data Byte						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xC7	CIRDAT_IN	7-0	RO	Received data to decode.	0x00	0xFE

CIR Counter Value for RLC Sample Period						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xC8	CIRRLC_CFG	7	R/W	Counter overflow control bit. **0**: if overflow, the counter will stop. **1**: if overflow, an interrupt issues and the counter keeps counting.	0x00	0xFE
		6-0	R/W	CIR RLC sample period, The unit is **1** μ s. Please note CIRRLC_CFG[6:0] can not be zero.		

CIR RLC Output 1st Byte						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xC9	CIRRLC_OUT0	7-0	R/W	Output (TX) 1st byte for RLC protocol.	0x00	0xFE

CIR RLC Output 2nd Byte						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xCA	CIRRLC_OUT1	7-0	R/W	Output (TX) 2nd byte for RLC protocol.	0x00	0xFE

CIR Carrier Discard/Average Pulse Number Setting for Automatic Carrier Detection.						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xCB	CIRCAR_PULS	7-4	R/W	Discard carrier pulse number F/W should specify the number of pulse to discard	0x44	0xFE
		3-0	R/W	Average carrier pulse number F/W should specify the average number to calculate the carrier period.		

CIR Detected Carrier Period						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xCC	CIRCAR_PRD	7	RO	Detected carrier period valid. **0**: carrier detection not completed. **1**: carrier detection completed.	0x00	0xFE
		6-0	RO	Detected carrier period. Detected carrier period = CIRCAR_PRD[6:0] x 500ns		

CIR Detected Pulse Width High of Carrier						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xCD	CIRCAR_HPRD	7	RSV	Reserved	0x00	0xFE
		6-0	R/W	Detected pulse width high of carrier Pulse width high = CIRCAR_HPRD[6:0] x 500ns		

CIR Modulation Carrier Period (RLC only)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xCE	CIRMOD_PRD	7	R/W	Carrier modulation selection. **0**: If TX idle state is high, Low signal in TX will be modulated. **1**: If TX idle state is low, High signal in TX will be modulated	0x00	0xFE
		6-0	R/W	Modulation carrier period. This register determines the modulation carrier period. The unit is 500ns. The value can be chosen from 0x02 to 0x7F, i.e., the period is from 15.87KHz~1MHz. The period = CIRMOD_PRD[6:0] x 500 ns.		

CIR Pulse Width High of Modulation Carrier (RLC only)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xCF	CIRMOD_HPRD	7	R/W	Reserved	0x00	0xFE
		6-0	R/W	Pulse width high of modulation carrier. This register determines the pulse width high of modulation carrier. The unit is 500ns. The value can be chosen from 0x01 to 0x7E. *Please note, the pulse width high can not be larger than the carrier period.* The pulse width high = CIRMOD_HPRD[6:0] x 500 ns.		

4.13.3 CIR Programming Sample

In this section gives some programming sample to control CIR module. Please note, ENE does not guarantee these codes in every field application. The following table describes scenario of CIR filed application.

Example
A RC6 receiver which filters out carrier is connected to CIR RX pin.
Programming model

GPIOIE40[0] (0xFC68[0]) = 1; //Enable CIR Rx input
CIRCFG (0xFEC0) = 0x07; //Enable Rx interrupt and protocol
CIRCFG2 (0xFEC1) = 0x02; //Select RC-6 protocol
CIRHIGH (0xFEC3) = 0x0B; //High width = 32*11 = 352 us
CIRBIT (0xFEC4) = 0x22; //Bit width = 32*34 = 1088 us
CIRSTART (0xFEC5) = 0x3B; //Leader width = 64*59 = 3776 us
CIRSTART2 (0xFEC6) = 0x4A; //Trailer width = 32*74 = 2368 us
When CIRPF[0] (0xFEC2[0]) = 1, Read CIRDAT_IN (0xFEC7) to get data.

4.14 PS/2 Interface (PS/2)

4.14.1 PS/2 Interface Function Description

The PS/2 protocol is a two-wire bi-direction interface in the industrial standard. This supports many PS/2 human interface devices, such as keyboard, mouse or touchpad device. Here gives the highlights of PS/2 features in the KBC.

- 3 external PS/2 channels supported.
- 1 internal PS/2 channel for IKB.
- Each PS/2 channel is with interrupt capability.
- Each PS/2 channel can be enabled/disabled individually.
- Both hardware and flexible firmware mode support for PS/2 protocol.
- Each PS/2 channel can be programmed to be GPIO function.

4.14.2 PS/2 Interface Registers Description

PS/2 Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xE0	PS2CFG	7	R/W	PS/2 port3 (TX/RX) enable. If disable, PS3CLK will be low. **0**: Disable **1**: Enable	0x00	0xFE
		6	R/W	PS/2 port2 (TX/RX) enable. If disable, PS2CLK will be low. **0**: Disable **1**: Enable		
		5	R/W	PS/2 port1 (TX/RX) enable. If disable, PS1CLK will be low. **0**: Disable **1**: Enable		
		4	R/W	PS/2 port0 IKB(TX/RX) enable. If disable, IKB clock will be low. **0**: Disable **1**: Enable		
		3	R/W	PS/2 parity error interrupt **0**: Disable **1**: Enable		
		2	R/W	PS/2 TX timeout interrupt. TX timeout condition: (a)ps2clk keeps high over 210 μs~240 μs during TX. (b)Host requests bus and waits over 120ms~150ms **0**: Disable **1**: Enable		
		1	R/W	PS/2 transmit-one-byte interrupt. **0**: Disable **1**: Enable		
		0	R/W	PS/2 receive-one-byte interrupt. **0**: Disable **1**: Enable		

PS/2 Interrupt Pending Flag						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xE1	PS2PF	7	RO	Flag of PS/2 port3 received one byte. **0**: no event **1**: event occurs	0x00	0xFE
		6	RO	Flag of PS/2 port2 received one byte. **0**: no event **1**: event occurs		
		5	RO	Flag of PS/2 port1 received one byte. **0**: no event **1**: event occurs		
		4	RO	Flag of PS/2 port0 (IKB) received one byte. **0**: no event **1**: event occurs		
		3	R/W1C	Interrupt flag of PS/2 parity error **0**: no event **1**: event occurs		
		2	R/W1C	Interrupt flag of PS/2 TX timeout. **0**: no event **1**: event occurs		
		1	R/W1C	Interrupt flag of PS/2 transmit-one-byte **0**: no event **1**: event occurs		
		0	R/W1C	Interrupt flag of PS/2 receive-one-byte **0**: no event **1**: event occurs		

PS/2 Transmitter/Receiver Control						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xE2	PS2CTRL	7	R/W	Data port **PS2DATA** (0xFEE3) connects to PS/2 port3 **0**: Disconnect **1**: Connect	0x00	0xFE
		6	R/W	Data port **PS2DATA** (0xFEE3) connects to PS/2 port2 **0**: Disconnect **1**: Connect		
		5	R/W	Data port **PS2DATA** (0xFEE3) connects to PS/2 port1 **0**: Disconnect **1**: Connect		
		4	R/W	Data port **PS2DATA** (0xFEE3) connects to PS/2 port0 **0**: Disconnect **1**: Connect		
		3	WO	Write "**1**" to force PS/2 TX reset.		
		2	WO	Write "**1**" to force PS/2 RX reset.		
		1	RO	PS/2 RX timeout flag. The flag may implies the followings. (a) ps2clk keeps high over 210 μs~240 μs during RX (b) host issues reset command and the device does not response. (c) General PS/2 packet timeout defined in the protocol.		
		0	RSV	Reserved		

PS/2 Data Port						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xE3	PS2DATA	7-0	R/W	EC F/W gets/writes data from/to host via this register.	0x00	0xFE

PS/2 Configuration 2						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xE4	PS2CFG2	7	R/W	PS/2 port3 hardware mode enable. **0**: Disable **1**: Enable	0x00	0xFE
		6	R/W	PS/2 port2 hardware mode enable. **0**: Disable **1**: Enable		
		5	R/W	PS/2 port1 hardware mode enable. **0**: Disable **1**: Enable		
		4	R/W	PS/2 port0 (IKB) hardware mode enable. **0**: Disable **1**: Enable		
		3	R/W	PS/2 hardware mode enable. **0**: Disable **1**: Enable		
		2	R/W	PS/2 host request timeout control. (in PS/2 hardware mode only) **0**: Host request timeout 120ms~150ms **1**: Host request timeout 15ms~16ms		
		1	RSV	Reserved.		
		0	R/W	PS/2 clock/data input debounce control **0**: 1 μs **1**: 2 μs		

PS/2 Pin Input Status						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xE5	PS2PINS	7	RO	PS/2 port3 clock pin status	0x00	0xFE
		6	RO	PS/2 port2 clock pin status		
		5	RO	PS/2 port1 clock pin status		
		4	RO	PS/2 port0 (IKB) clock pin status		
		3	RO	PS/2 port3 data pin status		
		2	RO	PS/2 port2 data pin status		
		1	RO	PS/2 port1 data pin status		
		0	RO	PS/2 port0 (IKB) data pin status		

PS/2 Pin Output						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xE6	PS2PINO	7	RO	PS/2 port3 clock pin status	0x00	0xFE
		6	RO	PS/2 port2 clock pin status		
		5	RO	PS/2 port1 clock pin status		
		4	RO	PS/2 port0 (IKB) clock pin status		
		3	RO	PS/2 port3 data pin status		
		2	RO	PS/2 port2 data pin status		
		1	RO	PS/2 port1 data pin status		
		0	RO	Reserved		

4.15 Embedded Controller (EC)

4.15.1 EC Function Description

The ACPI specification defined for the embedded controller (EC) interface requires either three separate host interfaces (KBC, SCI, SMI) or two interfaces (KBC, and shared SCI/SMI). The ENE KBC supports KBC and SCI interface, and SMI interface can be shared with SCI or use a dedicated GPIO. The embedded controller also provides some features which are collected as following:

- Handles EC standard commands from host, firmware mode support.
- Handles EC extended commands from host, only firmware mode support.
- SCI generation capability.
- Extended I/O write interface, i.e., debug port (port 80) support.
- KBC/EC clock configuration.
- A/D and D/A control.
- Power management control.
- Miscellaneous control.

The host queries (read) EC status and issues (write) EC command via port **66h**. The EC data port is **62h**. The status of EC is defined as the below table:

Status Bit	Name	Description
7	RSV	Reserved
6	RSV	Reserved
5	SCI	SCI event flag. Please note, this bit will not be set if standard EC commands (80h~84h) issued by host. **0:** No SCI event occurs **1:** SCI event occurs
4	Burst Enable	The burst enable flag **0:** Disable **1:** Enable
3	Command/Data Flag	**0:** Previous access port is data port. (EC_DAT) **1:** Previous access port is command/status port. (EC_CMD/EC_STS)
2	RSV	Reserved
1	IBF	**I**nput **B**uffer **F**ull flag of EC
0	OBF	**O**utput **B**uffer **F**ull flag of EC

The EC commands are defined as following, for more detail please refer to ACPI, *Advanced Configuration Power Interface Specification. 2.0*

Value	Command	Description
80h	EC Read	Read EC space registers
81h	EC Write	Write EC space registers
82h	EC Burst Enable	Enable EC operation in burst mode
83h	EC Burst Disable	Disable EC operation in burst mode
84h	EC Query	Query SCI events
Others	Firmware Command	Extended commands and handled with F/W mode.

4.15.2 EC Command Program Sequence

The following table summarizes the standard EC commands programming flow. Port **66h** is the EC command and status port and port **62h** is the EC data port.

Command Byte	Command	Program Sequence
80h	EC Read	1. Host writes command byte 80h (EC_Read) to port 66h. 2. EC will issue SCI to host while IBF=0 3. Host writes address to port 62h. 4. EC will issue SCI to host while OBF=1 5. Host reads data via port 62h.
81h	EC Write	1. Host writes command byte 81h (EC_Write) to port 66h. 2. EC will issue SCI to host while IBF=0 3. Host writes address to port 62h. 4. EC will issue SCI to host while IBF=0 5. Host writes data to port 62h. 6. EC will issue SCI to host while IBF=0
82h	Burst Enable	1. Host writes command byte 82h (Burst_Enable) to port 66h. 2. EC will issue SCI to host while OBF=1. 3. Host reads via port62h. If 90h obtained, it's Burst Ack.
83h	Burst Disable	1. Host writes command byte 83h (Burst_Disable) to port 66h. 2. EC will issue SCI to host while IBF=0
84h	Query EC	1. Host writes command byte 84h (Query_EC) to port 66h. 2. EC will issue SCI to host while OBF=1. 3. Host reads data via port 62h. The data obtained is SCI_ID number.

4.15.3 EC SCI Generation

The EC can generate SCI with independent enable control and status flag. Plenty of hardware SCI events are predefined, and a firmware SCI event gives more flexible use for different applications. There is a F/W SCI command port located at **SCID** (0xFF0B). As the F/W writes any **non-zero** value to this port, and corresponding enable bit (SCIE0[6]) is set. A hardware SCI signal will issue to host in sequence. Then the host uses standard EC_Query (84h) command to get the **SCI ID** which is written by F/W before. The below table summarizes the information about SCI events, SCI IDs and the priorities.

SCI ID	Event	Switch	Applications	Priority
00h	Nothing	N/A		0(Highest)
01h-07h	RSV	N/A	Reserved	1
08h	WDT	SCIE0[0]	Watchdog	2
09h	LPC_IO2F	SCIE0[1]	LPC I/O 0x2F R/W accessing interrupt	3
0Ah	PS2	SCIE0[2]	PS/2 event	4
0Bh	KBC	SCIE0[3]	IBF rising (LPC write I/O 60h/64h) OBF falling (LPC read I/O 60h)	5
0Ch	IKB	SCIE0[4]	IKB	6
0Dh	LPC_IO686C	SCIE0[5]	IBF rising (LPC write I/O 68h/6Ch) OBF falling (LPC read I/O 68h)	7
0Eh	LPC_IO6266	SCIE0[6]	IBF rising (LPC write I/O 62h/66h) OBF falling (LPC read I/O 62h)	8
FW_SCIID	*FW_SCI*	*SCIE0[7]*	*EC F/W SCI event*	9
10h	FAN0	SCIE1[0]	FAN0 monitor event (update/overflow)	10
11h	FAN1	SCIE1[1]	FAN1 monitor event (update/overflow)	11
12h	SMBus	SCIE1[2]	SMBus events	12
13h	CIR	SCIE1[3]	CIR events	13
14h	GPT0	SCIE1[4]	GPT0 event	14
15h	GPT1	SCIE1[5]	GPT1 event	15
16h	GPT2	SCIE1[6]	GPT2 event	16
17h	GPT3	SCIE1[7]	GPT3 event	17
18h	EXTWIO	SCIE3[0]	Write extended I/O (LPC I/O port 80)	18
19h	GPIO00~GPIO0F	SCIE3[1]	GPIO00~GPIO0F	19
1Ah	GPIO10~GPIO1F	SCIE3[2]	GPIO10~GPIO1F	20
1Bh	GPIO20~GPIO2F	SCIE3[3]	GPIO20~GPIO2F	21
1Ch	GPIO30~GPIO3F	SCIE3[4]	GPIO30~GPIO3F	22
1Dh	GPIO40~GPIO4F	SCIE3[5]	GPIO40~GPIO4F	23
1Eh	GPIO50~GPIO5F	SCIE3[6]	GPIO50~GPIO59 / GPXIOD00~GPXIOD07	24
1Fh	ADC	SCIE3[7]	ADC update	25(Lowest)

The SCI pulse width is programmable for different applications. Two unit basis, 16 μs and 64 μs can be chosen. To change the SCI pulse width, register **PXCFG**[2] (0xFF14) is for the unit selection and the width can be controlled via register **SCICFG**[3:0] (0xFF03). The equation shows the relationship. For more detail please refer to these 2 registers description.

*SCI Pulse Width = SCICFG[3:0] * Unit (16 μs or 64 μs)*

4.15.4 EC/KBC Clock Configuration

The EC provides programmable interface to adjust the microprocessor and peripheral frequency. By default, the microprocessor runs at 8MHz and peripherals are at 4MHz. The microprocessor can operate at the 32MHz as the highest frequency, and the peripheral runs up to 7.2 MHz. The programming interface is located at register **CLKCFG1/CLKCFG2** (0xFF0D/0xFF1E) and **PLLCFG/PLLCFG2** (0xFF0F/0xFF1F). The figure 4-1 illustrates the clock scheme applied in the KBC.

4.15.5 A/D Converter Control

The control interface of A/D is in the EC space. Features of the A/D converters are highlighted as following:

- Analog input range: *0.1*V_{CCA} ~0.9*V_{CCA}*
- INL: ±2 LSB (Typical)
- DNL: ± 1LSB (Typical)
- Resolution: 10-bit.
- Gain error: ± 1.5 LSB, Offset error: ± 1.5 LSB
- Six channels support.
- All A/D converters can be programmed to be *General Purpose Input* pins.

The following table summarizes the related registers of these 6 A/D converters.

Name	Address	Description
ADDAEN[3:0]	0xFF15	ADC Function Enable bits of ADC3~ADC0 Bit3: ADC3 Bit2: ADC2 Bit1: ADC1 Bit0: ADC0
ADCTRL[6:5]	0xFF18	ADC Function Enable bits of ADC5~ADC4 Bit6: ADC5 Bit5: ADC4
ADCTRL[4:2]	0xFF18	Select ADC channels to convert and output data in ADCDAT and ECIF[7:6]
ADCDAT	0xFF19	This stands for bit9~bit2 of 10bit A/D result.
ECIF[7:6]	0xFF1A	This stands for bit1~bit0 of 10bit A/D result.

The following gives the programming sample to control a ADC.

Example
Using ADC0 to get input analog signal
Programming model
6. set the related pin to be alternative output. GPIOIE38[0] (0xFC67[0]) = 1b 6. enable ADC function ADDAEN[6] (0xFF15[0]) = 1b 6. enable ADC control ADCTRL (0xFF18) = 0x01 Waiting ADC interrupr. 4. read ADCDAT (0xFF19) and ECIF (0xFF1A)

4.15.6 D/A Converter Control

The control interface of D/A is in the EC space. Features of the D/A converters are highlighted as following:

- Output range: AGND to VCCA
- Resolution: 8-bit.
- 4 channels support.
- All D/A converters are non-driving capability.
- All D/A converters can be programmed to be *General Purpose Output* pins.

The following table summarizes the related registers of these 4 D/A converters.

Name	Address	Description
ADDAEN[7:4]	0xFF15	DAC Function Enable bits of DAC3~DAC0 Bit7: DAC3 Bit6: DAC2 Bit5: DAC1 Bit4: DAC0 If DAC selected, please do not set related GPIO function selection register.
DAC0	0xFF10	DAC0 Output Value
DAC1	0xFF11	DAC1 Output Value
DAC2	0xFF12	DAC2 Output Value
DAC3	0xFF13	DAC3 Output Value

The following gives the programming sample to control a DAC.

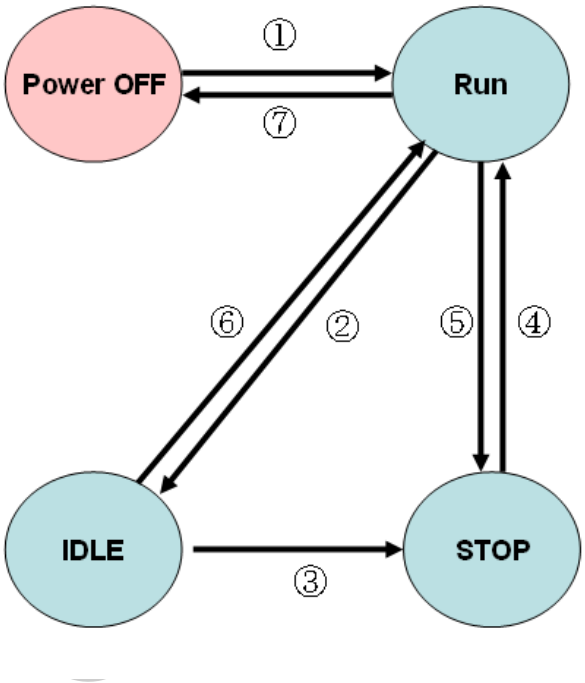
Example
Using DAC2
Programming model
6. set the related pin to be alternative output. GPIOFS38[6] (0xFC07[6]) = 0b 6. enable DAC function ADDAEN[6] (0xFF15[6]) = 1b 6. fill the value to be convert. DAC2 (0xFF12) = specific value to convert

4.15.7 Power Management Control

Two power modes are defined, one is **STOP** mode and the other is **IDLE** mode. The register **PMUCFG** (0xFF0C) is used to configure the power management. The following table gives more detail about the definition for these two power modes.

Mode	Description
STOP	All clock sources stop, except external PCI clock and 32.768KHz.
IDLE	Only clock of 8051 microprocessor stops.
RUN	System operations in normal mode.
OFF	All power supply removed, including AC and battery

The diagram below shows the relationship between each power mode.



① power supply on

② enter idle mode from run mode
set PMUCFG[6]=1

③ enter stop mode from idle mode
set PMUCFG[7]=1 (by Host)

④ back to run mode from stop mode
by wakeup source

⑤ enter stop mode from run mode
set PMUCFG[7]=1

⑥ back to run mode from idle mode
by wakeup source

⑦ all power removed

4.15.8 EC Registers Description

EC Hardware Revision ID						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x00	ECHV	7-0	RO	EC Hardware version	0xD2	0xFF

EC Firmware Revision ID						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x01	ECFV	7-0	R/W	EC firmware version This register will be a data port, **ADC_test_data**[7:0] in ADC test mode (ADCTR[1]=1).	0x00	0xFF

EC High Address						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x02	ECHA	7-6	R/W	These two bits will be a data port, **ADC_test_data**[9:8] in ADC test mode (ADCTR[1]=1).	0x0F	0xFF
		5	R/W	Write protection of PXCFG[1]. **0**: writable. **1**: write protection.		
		4	R/W	Index-I/O mode access control. **0**: access range 0xF400~0xFFFF **1**: access range 0xF400~0xF403 and 0xFC00~0xFFFF		
		3-0	RSV	Reserved		

EC SCI Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x03	SCICFG	7	R/W	Standard EC commands generate SCI. **0**: Disable **1**: Enable	0x90	0xFF
		6	R/W	**SCIID** port enable. (F/W SCI write port enable) **0**: Disable **1**: Enable		
		5	R/W	SCI polarity **0**: Low active (default) **1**: High active		
		4	R/W	SCIE0/SCIE1/SCIE2 (0xFF05~0xFF07) enable. **0**: Disable **1**: Enable		
		3-0	R/W	SCI pulse width. (max. 1ms) *SCI pulse width = SCICFG[3:0] * (time unit)* where time unit is determined by PXCFG[2], **64 μs or 16 μs** If SCICFG[3:0]=0, SCI pulse width = width of system clock.		

EC Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x04	ECCFG	7	R/W	EPB fast access enable. To enhance EPB performance. **0**: Disable **1**: Enable	0x00	0xFF
		6	R/W	Test mode selection **0**: Normal mode **1**: Test mode.		
		5-3	RSV	Reserved		
		2	R/W	Extended I/O (debug I/O, port 80) interrupt enable. Only available while write cycle to port 80 from the host. **0**: Disable **1**: Enable		
		1	R/W	IBF interrupt enable. EC command port interrupt enable. CPU writes command/data to EC command/data port. **0**: Disable **1**: Enable		
		0	R/W	OBF interrupt enable. EC data port interrupt enable. CPU reads data from EC data port. **0**: Disable **1**: Enable		

EC SCI Interrupt Enable (SCIE0,SCIE1,SCIE3)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x05	SCIE0	7-0	R/W	SCI Event0 enable **0**: Disable **1**: Enable	0x00	0xFF
0x06	SCIE1	7-0	R/W	SCI Event1 enable **0**: Disable **1**: Enable	0x00	0xFF
0x07	SCIE3	7-0	R/W	SCI Event2 enable **0**: Disable **1**: Enable	0x00	0xFF

EC SCI Flag (SCIF0,SCIF1,SCIF3)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x08	ECIF0	7-0	R/W1C	SCI Event0 flag **0**: no event **1**: event occurs	0x00	0xFF
0x09	ECIF1	7-0	R/W1C	SCI Event1 flag **0**: no event **1**: event occurs	0x00	0xFF
0x0A	ECIF3	7-0	R/W1C	SCI Event3 flag **0**: no event **1**: event occurs	0x00	0xFF

EC SCI ID Write Port (to Generate SCI Event)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x0B	SCID	7-0	R/W	Firmware SCI write port	0x00	0xFF

EC PMU Control/Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x0C	PMUCFG	7	WO	Write "**1**" to enter STOP mode.	0x2F	0xFF
		6	WO	Write "**1**" to enter Idle mode.		
		5	R/W	LPC cycle wakeup system from STOP mode. **0**: Disable **1**: Enable		
		4	R/W	Reset 8051 while in STOP mode. **0**: Disable **1**: Enable		
		3	R/W	SCI wakeup system **0**: Disable **1**: Enable		
		2	R/W	WDT wakeup system from STOP mode. **0**: Disable **1**: Enable		
		1	R/W	GPWU wakeup system from STOP mode. **0**: Disable **1**: Enable		
		0	R/W	Interrupt wakeup system from Idle mode. **0**: Disable **1**: Enable		

EC Clock Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x0D	CLKCFG	7	R/W	Flash clock from external clock (GPIO59). **0**: Disable **1**: Enable	0x00	0xFF
		6	R/W	Flash clock control. **0**: Half speed. (DPLL_CLK/2) **1**: Full speed (DPLL_CLK) please note, while CLKCFG[6]=0 and CLKCFG[3:2]=0 (power-on default), the SPI flash clock is always 16MHz.		
		5	R/W	DPLL generates 32.768 **MHz** **0**: Disable **1**: Enable		
		4	R/W	DPLL enters low power state while in STOP mode. **0**: Disable **1**: Enable		
		3-2	R/W	8051/Peripheral clock selection. **11b**: 32 MHz / 16 MHz **10b**: 22 MHz / 11 MHz **01b**: 16 MHz / 8 MHz **00b**: 8 MHz / 4 MHz (default)		
		1	R/W	Peripheral slow down to 1MHz automatically. If no host access, the peripheral clock will slow down to 1MHz automatically. **0**: Disable **1**: Enable		
		0	R/W	Clock slow down to 2MHz / 1MHz (8051 / Peripheral) in Idle mode. If this bit set, the clock of flash will be stopped in idle mode. **0**: Disable **1**: Enable		

EC Extended I/O (Debug Port) Write Data						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x0E	EXTIOW	7-0	R/W	If the host write data to extended I/O (debug port, port80), an interrupt occurs, and then the firmware read it back via this register.	0x00	0xFF

EC PLL Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x0F	PLLCFG	7-0	R/W	DPLL initial value. (low 8-bit) After reset, the DPLL will output frequency about 32MHz with default value 0xE0. DPLL initial value is 10-bit, the reset of two high bits are located at 0xFF1F, **PLLCFG2**[7:6].	0xE0	0xFF

EC DAC0 Output Value (ECMISC[1:0]=00b) / Extended Command (ECMISC[1:0]=11b)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x10	DAC0	7-0	R/W	The digital data to be converted in DAC0.	0x00	0xFF
0x10	EXTCMD	7-0	R/W	8051 extended command port. Once the command is filled, two events may occur. - if non-zero command written, 8051 interrupt issues. - If zero command written, SCI event issues. Please note, EXTARG0/EXTARG1/EXTARG2 must be ready before filling this register.	0x00	0xFF

EC DAC1 Output Value (ECMISC[1:0]=00b)/ Extended Command Argument 0(ECMISC[1:0]=11b)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x11	DAC1	7-0	R/W	The digital data to be converted in DAC1.	0x00	0xFF
0x11	EXTARG0	7-0	R/W	Extended command argument0	0x00	0xFF

EC DAC2 Output Value (ECMISC[1:0]=00b)/ Extended Command Argument 1(ECMISC[1:0]=11b)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x12	DAC2	7-0	R/W	The digital data to be converted in DAC2.	0x00	0xFF
0x12	EXTARG1	7-0	R/W	Extended command argument1	0x00	0xFF

EC DAC3 Output Value (ECMISC[1:0]=00b)/ Extended Command Argument 2(ECMISC[1:0]=11b)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x13	DAC3	7-0	R/W	The digital data to be converted in DAC3.	0x00	0xFF
0x13	EXTARG2	7-0	R/W	Extended command argument2	0x00	0xFF

EC 8051 On-Chip Control						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x14	PXCFG	7-3	RSV	Reserved	0x00	0xFF
		2	R/W	SCI pulse width time unit. **0**: 64 μ s **1**: 16 μ s		
		1	R/W	WDT timeout reset selection **0**: reset whole KBC, except GPIO module. **1**: reset 8051 only To write this bit, please make sure **ECHA**[5]=0.		
		0	R/W	8051 program counter control **0**: program counter starts to execute. **1**: 8051 reset and pc=0 (program counter). Pc will keep 0 (reset vector) until this bit is written to “0”		

EC ADC/DAC Function Switch						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x15	ADDAEN	7-4	R/W	DAC3~DAC0 Function Enable Bit7~Bit4 represents DAC3~DAC0 respectively **0:** DAC Disable **1:** DAC Enable If DAC enable, please **do not** set related GPIO function selection register.	0x00	0xFF
		3-0	R/W	ADC3~ADC0 Function Enable Bit3~Bit0 represents ADC3~ADC0 respectively **0:** ADC Disable **1:** ADC Enable. If ADC enable, please **do not** set related GPIO bit with input enable (IE).		

EC PLL Frequency Register (High Byte)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x16	PLLFRH	7-0	R/W	DPLL frequency = 32.768KHz(external) * PLLFR PLLFR[11:0] =(**PLLFRH**[7:0] : PLLFRL[7:4]) To generate 32.768MHz, PLLFR = 1000 (decimal) = 0x3E8 i.e., PLLFRH=0x3E	0x3E	0xFF

EC PLL Frequency Register (Low Byte)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x17	PLLFRL	7-4	R/W	DPLL frequency = 32.768KHz * PLLFR PLLFR[11:0] =(PLLFRH[7:0] : **PLLFRL**[7:4]) To generate 32.768MHz, PLLFR = 1000 (decimal) = 0x3E8 i.e., PLLFRL[7:4]=0x8	0x83	0xFF
		3	R/W	DPLL lock value presented in **CHIPID** (0xFF1E~0xFF1F). **0**: Disable **1**: Enable.		
		2	R/W	DPLL test mode enable **0**: Disable **1**: Enable.		
		1-0	RSV	Reserved		

EC ADC Control Register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x18	ADCTRL	7	RSV	Reserved	0x00	0xFF
		6-5	R/W	ADC5, ADC4 enable. Bit6 and Bit5 represent for ADC5 and ADC4 respectively. **0**: Disable **1**: Enable.		
		4-2	R/W	Convert ADC channel selection. **0**: ADC0 **1**: ADC1 **2**: ADC2 **3**: ADC3 **4**: ADC4 **5**: ADC5		
		1	R/W	ADC test mode enable. **0**: Disable **1**: Enable.		
		0	R/W	ADC convert start and force interrupt after converting. **0**: ADC stops converting, interrupt disable **1**: ADC starts converting, interrupt enable		

EC ADC Data Output Port						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x19	ADCDAT	7-0	RO	Converted data by ADC. ADC output[9:2]=**ADCDAT**[7:0]	0x00	0xFF

EC Interrupt Pending Flag						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x1A	ECIF	7-6	RO	Converted data by ADC. ADC output[1:0]=**ECIF**[7:6]	0x00	0xFF
		5-3	RSV	Reserved		
		2	R/W1C	EC firmware mode flag. If EC command handled by F/W, this flag will be set		
		1	R/W1C	EC IBF interrupt pending flag **0**: no event **1**: event occurs		
		0	R/W1C	EC OBF interrupt pending flag **0**: no event **1**: event occurs		

EC Data Port						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x1B	ECDAT	7-0	R/W	EC data port. If **ECDAT** written, **ECSTS**[0] (OBF) becomes “1”.	0x00	0xFF

EC Command Port						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x1C	ECCMD	7-0	RO	This register keeps EC command issued by the host.	0x00	0xFF

EC Control and Status Port						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x1D	ECSTS	7	R/W	Reserved	0x00	0xFF
		6	R/W	Reserved		
		5	RO	SCI pending flag **0**: no event **1**: event occurs		
		4	R/W	Burst enable status. **0**: EC burst mode disable **1**: EC burst mode enable.		
		3	R/W	EC I/O write port indicator **0**: host writes for data (writes I/O port 62h) **1**: host writes for command (writes I/O port 66h)		
		2	R/W	Register 0xFF1E and 0xFF1F function selection. **0**: CHIPID display selected **1**: CLKCFG2/PLLCFG2 function selected		
		1	R/W1C	IBF (Input Buffer Full) **0**: buffer not full **1**: buffer full		
		0	R/W1C	OBF (Output Buffer Full) **0**: buffer not full **1**: buffer full		

EC Clock Configuration 2						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x1E	CHIPID_H	7-0	R/W	**CHIPID** high byte. (ECSTS[2]=0)	0x39	0xFF
0x1E	CLKCFG2	7-0	R/W	Divider of **(DPLL Freq)/2** to generate 1 μs (ECSTS[2]=1) For example, DPLL outputs 64MHz (by default), to generate 1 μs, the divider should be 32. That is the CLKCFG2 will be 0x1F.	0x1F	0xFF

EC PLL Configuration 2						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x1F	CHIPID_L	7-0	R/W	**CHIPID** low byte. (ECSTS[2]=0)	0x26	0xFF
0x1F	PLLCFG2	7-6	R/W	High 2 bits of DPLL initial value. (ECSTS[2]=1) DPLL initial value is 10-bit, the low 8 bits are located at 0xFF0F, **PLLCFG**[7:0].	0x21	0xFF
		5	R/W	DPLL reference selection. **0**: PCI clock selected. (default) **1**: External 32.768KHz.		
		4	R/W	DPLL source clock divider. **0**: Disable **1**: Enable. (default). If PLLCFG2[5]=1, then this bit should be “1”. If PLLCFG2[5]=0, this bit should be “0”.		
		3-0	R/W	DPLL low speed state setting in Idle mode. The default value is **0001b**, the DPLL will provide 2MHz (8051)/ 1MHz (Peripheral) clock.		

EC MISC Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x20	ECMISC	7	R/W	8051 state. **0**: Idle state **1**: Normal state	0x00	0xFF
		6-3	R/W	Reserved		
		2	R/W	8051 extended command (**ExtCMD**, 0xFF10) interrupt enable. **0**: Disable **1**: Enable		
		1	R/W	Register function select of 0xFF10~0xFF13 for LPC index-I/O **0**: DAC **1**: 8051 Extended command related registers		
		0	R/W	Register function select of 0xFF10~0xFF13 for 8051. **0**: DAC **1**: 8051 Extended command related registers		

EC Extended I/O (Debug I/O) Data Port by Host						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x21	EXTIOR	7-0	R/W	The host reads extended I/O port and gets data from this register. *No interrupt occurs*.	0x00	0xFF

Embedded Debug Interface Feature Register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x22	EDIF	7	R/W	EDI feature enable **0**: disable **1**: enable	0x00	0xFF
		6-0	RSV	Reserved		

Embedded Debug Interface Active Status Register						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x23	EDIAS	7	R/W	EDI active status **0**: not active **1**: active	0x00	0xFF
		6-0	RSV	Reserved		

Embedded Debug Version ID						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x24	EDIID	7-0	RO	EDI version	0x01	0xFF

4.16 General Purpose Wake-up Controller (GPWU)

4.16.1 GPWU Function Description

The GPIO module provides flexible methods to wakeup the KBC or to generate interrupt. Once the input function is determined, plenty of features for wakeup can be setup. Here is the table to summarize all the features.

Wakeup Enable **0:** Disable **1:** Enable	**Polarity** **0:** ↓ / L **1:** ↑ / H	**Edge/Level** **0:** Edge **1:** Level	**Toggle** **0:** Disable **1:** Enable	**Description**
0	X	X	X	No wakeup events occur
1	X	X	1	Signal toggle trigger
1	0	0	0	Falling edge trigger
1	0	1	0	Low level trigger
1	1	0	0	Rising edge trigger
1	1	1	0	High level trigger

4.16.2 GPWU Registers Description

GPIO Wakeup Event Enable						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x30	GPWUEN00	7-0	R/W	GPIO00~GPIO07 Wakeup Event Switch bit[0]~bit[7] stand for GPIO00~GPIO07 separately **0:** Wakeup event disable **1**:Wakeup event enable	0x00	0xFF
0x31	GPWUEN08	7-0	R/W	GPIO08~GPIO0F Wakeup Event Switch bit[0]~bit[7] stand for GPIO08~GPIO0F separately **0:** Wakeup event disable **1**:Wakeup event enable	0x00	0xFF
0x32	GPWUEN10	7-0	R/W	GPIO10~GPIO17 Wakeup Event Switch bit[0]~bit[7] stand for GPIO10~GPIO17 separately **0:** Wakeup event disable **1**:Wakeup event enable	0x00	0xFF
0x33	GPWUEN18	7-0	R/W	GPIO18~GPIO1F Wakeup Event Switch bit[0]~bit[7] stand for GPIO18~GPIO1F separately **0:** Wakeup event disable **1**:Wakeup event enable	0x00	0xFF
0x34	GPWUEN20	7-0	R/W	GPIO20~GPIO27 Wakeup Event Switch bit[0]~bit[7] stand for GPIO20~GPIO27 separately **0:** Wakeup event disable **1**:Wakeup event enable	0x00	0xFF
0x35	GPWUEN28	7-0	R/W	GPIO28~GPIO2F Wakeup Event Switch bit[0]~bit[7] stand for GPIO28~GPIO2F separately **0:** Wakeup event disable **1**:Wakeup event enable	0x00	0xFF
0x36	GPWUEN30	7-0	R/W	GPIO30~GPIO37 Wakeup Event Switch bit[0]~bit[7] stand for GPIO30~GPIO37 separately **0:** Wakeup event disable **1**:Wakeup event enable	0x00	0xFF
0x37	GPWUEN38	7-0	R/W	GPIO38~GPIO3B Wakeup Event Switch bit[0]~bit[3] stand for GPIO38~GPIO3B separately **0:** Wakeup event disable **1**:Wakeup event enable ** GPO3C~GPO3F have no input functions.*	0x00	0xFF
0x38	GPWUEN40	7-0	R/W	GPIO40~GPIO47 Wakeup Event Switch bit[0]~bit[7] stand for GPIO40~GPIO47 separately **0:** Wakeup event disable **1**:Wakeup event enable	0x00	0xFF
0x39	GPWUEN48	7-0	R/W	GPIO48~GPIO4F Wakeup Event Switch bit[0]~bit[7] stand for GPIO48~GPIO4F separately **0:** Wakeup event disable **1**:Wakeup event enable	0x00	0xFF

0x3A	GPWUEN50	7-0	R/W	GPIO50~GPIO57 Wakeup Event Switch bit[0]~bit[7] stand for GPIO50~GPIO57 separately **0:** Wakeup event disable **1**:Wakeup event enable	0x00	0xFF
0x3B	GPWUEN58	7-0	R/W	GPIO58~GPIO59 Wakeup Event Switch bit[0]~bit[1] stand for GPIO58~GPIO59 separately **0:** Wakeup event disable **1**:Wakeup event enable	0x00	0xFF
0x3C	GPWUEN60	7-0	R/W	GPXIOD00~GPXIOD07 Wakeup Event Switch bit[0]~bit[1] stand for GPXIOD00~GPXIOD07 separately **0:** Wakeup event disable **1**:Wakeup event enable	0x00	0xFF

GPIO Wakeup Event Pending Flag						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x40	GPWUPF00	7-0	R/W1C	GPIO00~GPIO07 Wakeup Event Pending Flag bit[0]~bit[7] stand for GPIO00~GPIO07 separately **0:** No wakeup event **1**:Wakeup event pending	0x00	0xFF
0x41	GPWUPF08	7-0	R/W1C	GPIO08~GPIO0F Wakeup Event Pending Flag bit[0]~bit[7] stand for GPIO08~GPIO0F separately **0:** No wakeup event **1**:Wakeup event pending	0x00	0xFF
0x42	GPWUPF10	7-0	R/W1C	GPIO10~GPIO17 Wakeup Event Pending Flag bit[0]~bit[7] stand for GPIO10~GPIO17 separately **0:** No wakeup event **1**:Wakeup event pending	0x00	0xFF
0x43	GPWUPF18	7-0	R/W1C	GPIO18~GPIO1F Wakeup Event Pending Flag bit[0]~bit[7] stand for GPIO18~GPIO1F separately **0:** No wakeup event **1**:Wakeup event pending	0x00	0xFF
0x44	GPWUPF20	7-0	R/W1C	GPIO20~GPIO27 Wakeup Event Pending Flag bit[0]~bit[7] stand for GPIO20~GPIO27 separately **0:** No wakeup event **1**:Wakeup event pending	0x00	0xFF
0x45	GPWUPF28	7-0	R/W1C	GPIO28~GPIO2F Wakeup Event Pending Flag bit[0]~bit[7] stand for GPIO28~GPIO2F separately **0:** No wakeup event **1**:Wakeup event pending	0x00	0xFF
0x46	GPWUPF30	7-0	R/W1C	GPIO30~GPIO37 Wakeup Event Pending Flag bit[0]~bit[7] stand for GPIO30~GPIO37 separately **0:** No wakeup event **1**:Wakeup event pending	0x00	0xFF
0x47	GPWUPF38	7-0	R/W1C	GPIO38~GPIO3B Wakeup Event Pending Flag bit[0]~bit[3] stand for GPIO38~GPIO3B separately **0:** No wakeup event **1**:Wakeup event pending ** GPO3C~GPO3F have no input functions.*	0x00	0xFF
0x48	GPWUPF40	7-0	R/W1C	GPIO40~GPIO47 Wakeup Event Pending Flag bit[0]~bit[7] stand for GPIO40~GPIO47 separately **0:** No wakeup event **1**:Wakeup event pending	0x00	0xFF
0x49	GPWUPF48	7-0	R/W1C	GPIO48~GPIO4F Wakeup Event Pending Flag bit[0]~bit[7] stand for GPIO48~GPIO4F separately **0:** No wakeup event **1**:Wakeup event pending	0x00	0xFF
0x4A	GPWUPf50	7-0	R/W1C	GPIO50~GPIO57 Wakeup Event Pending Flag bit[0]~bit[7] stand for GPIO50~GPIO57 separately **0:** No wakeup event **1**:Wakeup event pending	0x00	0xFF
0x4B	GPWUPF58	7-0	R/W1C	GPIO58~GPIO59 Wakeup Event Pending Flag bit[0]~bit[1] stand for GPIO58~GPIO59 separately **0:** No wakeup event **1**:Wakeup event pending	0x00	0xFF
0x4C	GXWUPF00	7-0	R/W1C	GPXIOD00~GPXIOD07 Wakeup Event Pending Flag bit[0]~bit[1] stand for GPXIOD00~GPXIOD07 separately **0:** No wakeup event **1:**Wakeup event pending		

GPIO Wakeup Polarity Selection						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x50	GPWUPS00	7-0	R/W	GPIO00~GPIO07 Wakeup Polarity Selection bit[0]~bit[7] stand for GPIO00~GPIO07 separately **0:** Low active (level trigger) / Falling (edge trigger) **1:**High active (high trigger) / Rising (edge trigger)	0x00	0xFF
0x51	GPWUPS08	7-0	R/W	GPIO08~GPIO0F Wakeup Polarity Selection bit[0]~bit[7] stand for GPIO08~GPIO0F separately **0:** Low active (level trigger) / Falling (edge trigger) **1:**High active (high trigger) / Rising (edge trigger)	0x00	0xFF
0x52	GPWUPS10	7-0	R/W	GPIO10~GPIO17 Wakeup Polarity Selection bit[0]~bit[7] stand for GPIO10~GPIO17 separately **0:** Low active (level trigger) / Falling (edge trigger) **1:**High active (high trigger) / Rising (edge trigger)	0x00	0xFF
0x53	GPWUPS18	7-0	R/W	GPIO18~GPIO1F Wakeup Polarity Selection bit[0]~bit[7] stand for GPIO18~GPIO1F separately **0:** Low active (level trigger) / Falling (edge trigger) **1:**High active (high trigger) / Rising (edge trigger)	0x00	0xFF
0x54	GPWUPS20	7-0	R/W	GPIO20~GPIO27 Wakeup Polarity Selection bit[0]~bit[7] stand for GPIO20~GPIO27 separately **0:** Low active (level trigger) / Falling (edge trigger) **1:**High active (high trigger) / Rising (edge trigger)	0x00	0xFF
0x55	GPWUPS28	7-0	R/W	GPIO28~GPIO2F Wakeup Polarity Selection bit[0]~bit[7] stand for GPIO28~GPIO2F separately **0:** Low active (level trigger) / Falling (edge trigger) **1:**High active (high trigger) / Rising (edge trigger)	0x00	0xFF
0x56	GPWUPS30	7-0	R/W	GPIO30~GPIO37 Wakeup Polarity Selection bit[0]~bit[7] stand for GPIO30~GPIO37 separately **0:** Low active (level trigger) / Falling (edge trigger) **1:**High active (high trigger) / Rising (edge trigger)	0x00	0xFF
0x57	GPWUPS38	7-0	R/W	GPIO38~GPIO3B Wakeup Polarity Selection bit[0]~bit[3] stand for GPIO38~GPIO3B separately **0:** Low active (level trigger) / Falling (edge trigger) **1:**High active (high trigger) / Rising (edge trigger) ** GPO3C~GPO3F have no input functions.*	0x00	0xFF
0x58	GPWUPS40	7-0	R/W	GPIO40~GPIO47 Wakeup Polarity Selection bit[0]~bit[7] stand for GPIO40~GPIO47 separately **0:** Low active (level trigger) / Falling (edge trigger) **1:**High active (high trigger) / Rising (edge trigger)	0x00	0xFF
0x59	GPWUPS48	7-0	R/W	GPIO48~GPIO4F Wakeup Polarity Selection bit[0]~bit[7] stand for GPIO48~GPIO4F separately **0:** Low active (level trigger) / Falling (edge trigger) **1:**High active (high trigger) / Rising (edge trigger)	0x00	0xFF
0x5A	GPWUPS50	7-0	R/W	GPIO50~GPIO57 Wakeup Polarity Selection bit[0]~bit[7] stand for GPIO50~GPIO57 separately **0:** Low active (level trigger) / Falling (edge trigger) **1:**High active (high trigger) / Rising (edge trigger)	0x00	0xFF
0x5B	GPWUPS58	7-0	R/W	GPIO58~GPIO59 Wakeup Polarity Selection bit[0]~bit[1] stand for GPIO58~GPIO59 separately **0:** Low active (level trigger) / Falling (edge trigger) **1:**High active (high trigger) / Rising (edge trigger)	0x00	0xFF
0x5C	GXWUPS00	7-0	R/W	GPXIOD00~GPXIOD07 Wakeup Polarity Selection bit[0]~bit[1] stand for GPXIOD00~GPXIOD07 separately **0:** Low active (level trigger) / Falling (edge trigger) **1:**High active (high trigger) / Rising (edge trigger)		

GPIO Wakeup Level/Edge Trigger Selection						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x60	GPWUEL00	7-0	R/W	GPIO00~GPIO07 Wakeup Level/Edge Selection bit[0]~bit[7] stand for GPIO00~GPIO07 separately **0:** Edge trigger **1:** Level trigger	0x00	0xFF
0x61	GPWUEL08	7-0	R/W	GPIO08~GPIO0F Wakeup Level/Edge Selection bit[0]~bit[7] stand for GPIO08~GPIO0F separately **0:** Edge trigger **1:** Level trigger	0x00	0xFF
0x62	GPWUEL10	7-0	R/W	GPIO10~GPIO17 Wakeup Level/Edge Selection bit[0]~bit[7] stand for GPIO10~GPIO17 separately **0:** Edge trigger **1:** Level trigger	0x00	0xFF
0x63	GPWUEL18	7-0	R/W	GPIO18~GPIO1F Wakeup Level/Edge Selection bit[0]~bit[7] stand for GPIO18~GPIO1F separately **0:** Edge trigger **1:** Level trigger	0x00	0xFF
0x64	GPWUEL20	7-0	R/W	GPIO20~GPIO27 Wakeup Level/Edge Selection bit[0]~bit[7] stand for GPIO20~GPIO27 separately **0:** Edge trigger **1:** Level trigger	0x00	0xFF
0x65	GPWUEL28	7-0	R/W	GPIO28~GPIO2F Wakeup Level/Edge Selection bit[0]~bit[7] stand for GPIO28~GPIO2F separately **0:** Edge trigger **1:** Level trigger	0x00	0xFF
0x66	GPWUEL30	7-0	R/W	GPIO30~GPIO37 Wakeup Level/Edge Selection bit[0]~bit[7] stand for GPIO30~GPIO37 separately **0:** Edge trigger **1:** Level trigger	0x00	0xFF
0x67	GPWUEL38	7-0	R/W	GPIO38~GPIO3B Wakeup Level/Edge Selection bit[0]~bit[3] stand for GPIO38~GPIO3B separately **0:** Edge trigger **1:** Level trigger ** GPO3C~GPO3F have no input functions.*	0x00	0xFF
0x68	GPWUEL40	7-0	R/W	GPIO40~GPIO47 Wakeup Level/Edge Selection bit[0]~bit[7] stand for GPIO40~GPIO47 separately **0:** Edge trigger **1:** Level trigger	0x00	0xFF
0x69	GPWUEL48	7-0	R/W	GPIO48~GPIO4F Wakeup Level/Edge Selection bit[0]~bit[7] stand for GPIO48~GPIO4F separately **0:** Edge trigger **1:** Level trigger	0x00	0xFF
0x6A	GPWUEL50	7-0	R/W	GPIO50~GPIO57 Wakeup Level/Edge Selection bit[0]~bit[7] stand for GPIO50~GPIO57 separately **0:** Edge trigger **1:** Level trigger	0x00	0xFF
0x6B	GPWUEL58	7-0	R/W	GPIO58~GPIO59 Wakeup Level/Edge Selection bit[0]~bit[1] stand for GPIO58~GPIO59 separately **0:** Edge trigger **1:** Level trigger	0x00	0xFF
0x6C	GXWUEL00	7-0	R/W	GPXIOD00~GPXIOD07 Wakeup Level/Edge Selection bit[0]~bit[1] stand for GPXIOD00~GPXIOD07 separately **0:** Edge trigger **1:** Level trigger		

GPIO Wakeup Input Change (Toggle) Trigger Selection						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x70	GPWUCHG00	7-0	R/W	GPIO00~GPIO07 Wakeup Input Change (Toggle) Trigger bit[0]~bit[7] stand for GPIO00~GPIO07 separately This setting will ignore the corresponding bit of GPWUELxx. **0:** Toggle trigger disable **1:** Toggle trigger enable	0x00	0xFF
0x71	GPWUCHG08	7-0	R/W	GPIO08~GPIO0F Wakeup Input Change (Toggle) Trigger bit[0]~bit[7] stand for GPIO08~GPIO0F separately This setting will ignore the corresponding bit of GPWUELxx. **0:** Toggle trigger disable **1:** Toggle trigger enable	0x00	0xFF
0x72	GPWUCHG10	7-0	R/W	GPIO10~GPIO17 Wakeup Input Change (Toggle) Trigger bit[0]~bit[7] stand for GPIO10~GPIO17 separately This setting will ignore the corresponding bit of GPWUELxx. **0:** Toggle trigger disable **1:** Toggle trigger enable	0x00	0xFF
0x73	GPWUCHG18	7-0	R/W	GPIO18~GPIO1F Wakeup Input Change (Toggle) Trigger bit[0]~bit[7] stand for GPIO18~GPIO1F separately This setting will ignore the corresponding bit of GPWUELxx. **0:** Toggle trigger disable **1:** Toggle trigger enable	0x00	0xFF
0x74	GPWUCHG20	7-0	R/W	GPIO20~GPIO27 Wakeup Input Change (Toggle) Trigger bit[0]~bit[7] stand for GPIO20~GPIO27 separately This setting will ignore the corresponding bit of GPWUELxx. **0:** Toggle trigger disable **1:** Toggle trigger enable	0x00	0xFF
0x75	GPWUCHG28	7-0	R/W	GPIO28~GPIO2F Wakeup Input Change (Toggle) Trigger bit[0]~bit[7] stand for GPIO28~GPIO2F separately This setting will ignore the corresponding bit of GPWUELxx. **0:** Toggle trigger disable **1:** Toggle trigger enable	0x00	0xFF
0x76	GPWUCHG30	7-0	R/W	GPIO30~GPIO37 Wakeup Input Change (Toggle) Trigger bit[0]~bit[7] stand for GPIO30~GPIO37 separately This setting will ignore the corresponding bit of GPWUELxx. **0:** Toggle trigger disable **1:** Toggle trigger enable	0x00	0xFF
0x77	GPWUCHG38	7-0	R/W	GPIO38~GPIO3B Wakeup Input Change (Toggle) Trigger bit[0]~bit[3] stand for GPIO38~GPIO3B separately This setting will ignore the corresponding bit of GPWUELxx. **0:** Toggle trigger disable **1:** Toggle trigger enable ** GPO3C~GPO3F have no input functions.*	0x00	0xFF

0x78	GPWUCHG40	7-0	R/W	GPIO40~GPIO47 Wakeup Input Change (Toggle) Trigger bit[0]~bit[7] stand for GPIO40~GPIO47 separately This setting will ignore the corresponding bit of GPWUELxx. **0:** Toggle trigger disable **1:** Toggle trigger enable	0x00	0xFF
0x79	GPWUCHG48	7-0	R/W	GPIO48~GPIO4F Wakeup Input Change (Toggle) Trigger bit[0]~bit[7] stand for GPIO48~GPIO4F separately This setting will ignore the corresponding bit of GPWUELxx. **0:** Toggle trigger disable **1:** Toggle trigger enable	0x00	0xFF
0x7A	GPWUCHG50	7-0	R/W	GPIO50~GPIO57 Wakeup Input Change (Toggle) Trigger bit[0]~bit[7] stand for GPIO50~GPIO57 separately This setting will ignore the corresponding bit of GPWUELxx. **0:** Toggle trigger disable **1:** Toggle trigger enable	0x00	0xFF
0x7B	GPWUCHG58	7-0	R/W	GPIO58~GPIO59 Wakeup Input Change (Toggle) Trigger bit[0]~bit[1] stand for GPIO58~GPIO59 separately This setting will ignore the corresponding bit of GPWUELxx. **0:** Toggle trigger disable **1:** Toggle trigger enable	0x00	0xFF
0x7C	GXWUCHG00	7-0	R/W	GPXIOD00~GPXIOD07 Wakeup Input Change (Toggle) Trigger bit[0]~bit[1] stand for GPXIOD00~GPXIOD07 separately This setting will ignore the corresponding bit of GPWUELxx. **0:** Toggle trigger disable **1:** Toggle trigger enable		

4.16.3 GPWU Programming Sample

In this section gives some programming sample to control GPWU module. Please note, ENE does not guarantee these codes in every field application. The following table describes scenario of GPWU filed application.

Example	
PIN	**Function**
GPIO02	Low level trigger
GPIO03	Rising edge trigger
GPIO05	Falling edge trigger
GPIO06	Edge change trigger
Programming model	
6. set related wakeup enable register. GPIWUEN00 (0xFF30) = 0x6C 6. set related wakeup polarity register GPWUPS00 (0xFF50) = 0x08 6. set related wakeup edge/level trigger register GPWUEL00 (0xFC60) = 0x04 4. set related wakeup input change register GPWUCHG00 (0xFF70) = 0x40	

4.17 System Management Bus Controller (SMBus)

4.17.1 SMBus Function Description

The SMBus is a two wire interface design based on I^2C bus. The SMBus controller in the KBC supports SMBus 2.0 and supports both master and slave mode with 2 channels. The SMBus controller supports 12 command protocols as following table. For more detail about each command protocol, please refer to the *System Management Bus Specification 2.0.*

Command Byte	Command	Command Byte	Command
02h	Quick Write	08h	Write Word
03h	Quick Read	09h	Read Word
04h	Send Byte	0Ah	Write Block
05h	Receive Byte	0Bh	Read Block
06h	Write Byte	0Ch	Word Process
07h	Read Byte	0Dh	Block Process

The SMBus introduces new mechanism to communicate with I^2C devices, called **Byte mode**. If the SMBus operates in this mode, only 3 protocols are supported, ***05h*** *(Receive Byte)*, ***0Ah*** *(Write Block)* and ***0Bh*** *(Read Block)*. Here gives the brief programming guide of how to use Byte mode as following table.

05h, Receive Byte	0Ah, Write Block	0Bh, Read Block
1. Set the address in SMBADR (0xFF9A). 2. Set the ACK or NACK bit in SMBPF (0xFF96[6]). 3. Set the protocol in SMBPRTCL (0xFF98). 4. Once one byte data received, the interrupt pending flag will be set (0xFF96[5]). And the F/W could obtain the data via pooling or interrupt method. 5. If more than one byte received, the F/W must set the ACK or NACK response in advance. (the same as step 2), then continue to the step 4 until all bytes complete.	1. Set the address in SMBADR (0xFF9A). 2. Set the data array in SMBDAT (0xFF9C). 3. Set the count number in SMBCNT (0xFFBC). 4. Set the protocol in SMBPRTCL (0xFF98).	1. Set the address in SMBADR (0xFF9A). 2. Set the count number in SMBCNT (0xFFBC). 3. Set the protocol in SMBPRTCL (0xFF98).

The SMBus controller works as a host (master). The controller can be programmed to enable slave mode. In slave mode, the controller will response to its slave address which is programmable. A slave device could communicate with the SMBus host controller via **SMBus Alert** or **Host Notify** protocols. The **SMBus Alert** protocol can be implemented via optional SMBAlert# signal or periodical ARA (Alert Response Address) command. As to **Host Notify** protocol, The controller provides registers for F/W to achieve different applications. The following gives the brief summary between Host Notify protocol and SMBus register interface.

1bit	7bit	1bit	1bit	7bit	1bit	8bit	1bit	8bit	1bit	1bit
S	SMB Host Addr.	Wr	A	Device Addr.	A	Data Low Byte	A	Data High Byte	A	P

SMB Host Addr : stored in **SMBAADR**, 0xFFBD.
Device Addr : stored in **SMBAADR**, 0xFFBD.
Data Low Byte: stored in **SMBADAT0**, 0xFFBE.
Data High Byte: stored in **SMBADAT1**, 0xFFBF.
S: Start bit P: Stop bit

Slave (SMBus device) to Master

Master (SMBus host) to Slave

SMBus Register Description

SMBus CRC Value						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x92	SMBTCRC	7-0	RO	SMBus CRC value.	0x00	0xFF

SMBus Pin Control						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x93	SMBPIN	7	R/W	SMBus data line forced to low. Write "**0**" to force **SDA0** or **SDA1** low.	0x00	0xFF
		6	R/W	SMBus clock line forced to low. Write "**0**" to force **SCL0** or **SCL1** low.		
		5	RO	Status of SDA0 or SDA1 or SDA0 wired SDA1..		
		4	RO	Status of SCL0 or SCL1 or SCL0 wired SCL1.		
		3	R/W	Byte mode function enable 3 protocols support, **Write Block/Read Block/Receive Byte.** Protocols are defined via register SMBPRTCL[6:0] **0**: Disable **1**: Enable		
		2	R/W	SCL/SDA input debounce enable. **0**: Disable **1**: Enable		
		1	R/W	SCL1/SDA1 pin connected to SMBus controller. **0**: Disable **1**: Enable		
		0	R/W	SCL0/SDA0 pin connected to SMBus controller. **0**: Disable **1**: Enable		

SMBus Configuration						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x94	SMBCFG	7	R/W	SMBus master disable **0**: Enable master function. **1**: Disable master function	0x06	0xFF
		6	R/W	SMBus host alarm protocol disable (0xFFBD~0xFFBF disable) **0**: Enable slave function. **1**: Disable slave function		
		5	RSV	Reserved		
		4-0	R/W	SMBus clock period If **SMBCFG[4:0]>0 and SMBPIN[2]=1,** the period is SMBus clock period = (SMBCFG[4:0]+1) * 4 μs If **SMBCFG[4:0]>0 and SMBPIN[2]=0**, the period is SMBus clock period = SMBCFG[4:0] * 4 μs Please **do not** set these bits to "0".		

SMBus Interrupt Enable						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x95	SMBEN	7	RO	SMBus host controller status **0**: not busy **1**: busy	0x00	0xFF
		6-4	RSV	Reserved		
		3	R/W	SMBus slave protocol selection. **0**: word read/write **1**: byte read/write		
		2	R/W	SMBus slave mode enable. **0**: Disable **1**: Enable		
		1	R/W	SMBus alert (host notify protocol) interrupt **0**: Disable **1**: Enable		
		0	R/W	SMBus protocol completion interrupt **0**: Disable **1**: Enable		

SMBus Interrupt Pending Flag						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x96	SMBPF	7	RSV	Reserved	0x00	0xFF
		6	R/W	ACK bit of Receive Byte (Byte Mode) protocol **0**: ACK, the Receive Byte protocol keeps going **1**: NACK, once the F/W ready to obtain the last Receive Byte, F/W set this bit in advance. After this last byte transferred, the controller issues NACK to device and the protocol stop.		
		5	R/W1C	Read data interrupt flag of Receive Byte (Byte Mode) protocol **0**: no event **1**: event occurs		
		4	RO	Read protocol interrupt flag of SMBus slave **0**: no event **1**: event occurs		
		3	R/W1C	Interrupt flag of SMBus slave **0**: no event **1**: event occurs		
		2-0	RSV	Reserved		

SMBus Received CRC Value						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x97	SMBRCRC	7-0	RO	The CRC value received from SMBus slave device.	0x00	0xFF

SMBus Protocol						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x98	SMBPRTCL	7	R/W	SMBus transaction with PEC (Packet Error Check) **0**: Disable **1**: Enable.	0x00	0xFF
		6-0	R/W	Command protocol. **02h**: Quick Write **03h**: Quick Read **04h**: Send Byte **05h**: Receive Byte / Receive Byte (Byte Mode) **06h**: Write Byte **07h**: Read Byte **08h**: Write Word **09h**: Read Word **0Ah**: Write Block **/** Write Block (Byte Mode) **0Bh**: Write Read / Read Block (Byte Mode) **0Ch**: Word Process **0Dh**: Block Process others: Reserved		

SMBus Status						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x99	SMBSTS	7	R/W	SMBus command done flag **0**: no event (Write 0 to clear) **1**: event occurs	0x00	0xFF
		6	R/W	SMBus alarm (host notify protocol) interrupt flag **0**: no event (Write 0 to clear) **1**: event occurs		
		5	R/W	SMBus block data array protocol control. F/W could control the protocol progress via this bit. **0**: F/W clear this bit to continue un-finished SMBUS block transaction **1**: H/W automatically set this bit when SMBUS block transaction is un-finished (SMBUS clock is driven to low), wait F/W clear this bit to continue this un-finished SMBUS block transaction.		
		4-0	R/W	Error code. **00h**: no error **07h**: unknown address failure. **10h**: device address no ACK **12h**: command no ACK **13h**: device data no ACK **17h**: device access deny **18h**: SMBus timeout **19h:** unsupported protocol **1Ah**: SMBus busy **1Fh**: PEC (Packet Error Check) error others: Reserved		

SMBus Address Port						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x9A	SMBADR	7-0	R/W	SMBus address (7-bits long), bit0 ignored.	0x00	0xFF
0x9A	SMBADR (SMBPIN[3]=1)	7-1	R/W	SMBus address (7-bits long).		
		0	R/W	Data direction bit **0**: Write **1**: Read		

SMBus Command Port						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x9B	SMBCMD	7-0	R/W	SMBus command port	0x00	0xFF

SMBus Data Array (8 Bytes)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0x9C	SMBDAT0	7-0	R/W	Data port for ***Send/Receive/Read Byte/Write Byte*** protocol	0x00	0xFF
0x9D	SMBDAT1	7-0	R/W	Data port for ***Read Word/Write Word*** protocol	0x00	0xFF
0x9E	SMBDAT2	7-0	R/W	Data port for ***Block*** protocol	0x00	0xFF
0x9F	SMBDAT3	7-0	R/W	Data port for ***Block*** protocol	0x00	0xFF
0xA0	SMBDAT4	7-0	R/W	Data port for ***Block*** protocol	0x00	0xFF
0xA1	SMBDAT5	7-0	R/W	Data port for ***Block*** protocol	0x00	0xFF
0xA2	SMBDAT6	7-0	R/W	Data port for ***Block*** protocol	0x00	0xFF
0xA3	SMBDAT7	7-0	R/W	Data port for ***Block*** protocol	0x00	0xFF

SMBus Slave Address (SMBEN[2]=1)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xA4	SMBRSA	7-0	R/W	SMBus slave address (7-bits long), bit0 ignores.	0x00	0xFF

SMBus Block Count						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xBC	SMBCNT	7-0	R/W	Smbus block count. If “0x00”, it means 32-byte length in a block transfer. Bit7~Bit5 are ignored	0x00	0xFF

SMBus Alarm (Host Notify Protocol) Address						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xBD	SMBAADR	7-0	R/W	This register is alarm address.	0x00	0xFF

SMBus Alarm Data 0 (Low Byte)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xBE	SMBDAT0	7-0	R/W	Alarm data (low byte)	0x00	0xFF

SMBus Alarm Data 1 (High Byte)						
Offset	**Name**	**Bit**	**Type**	**Description**	**Default**	**Bank**
0xBF	SMBDAT1	7-0	R/W	Alarm data (high byte)	0x00	0xFF

SMBus Programming Sample

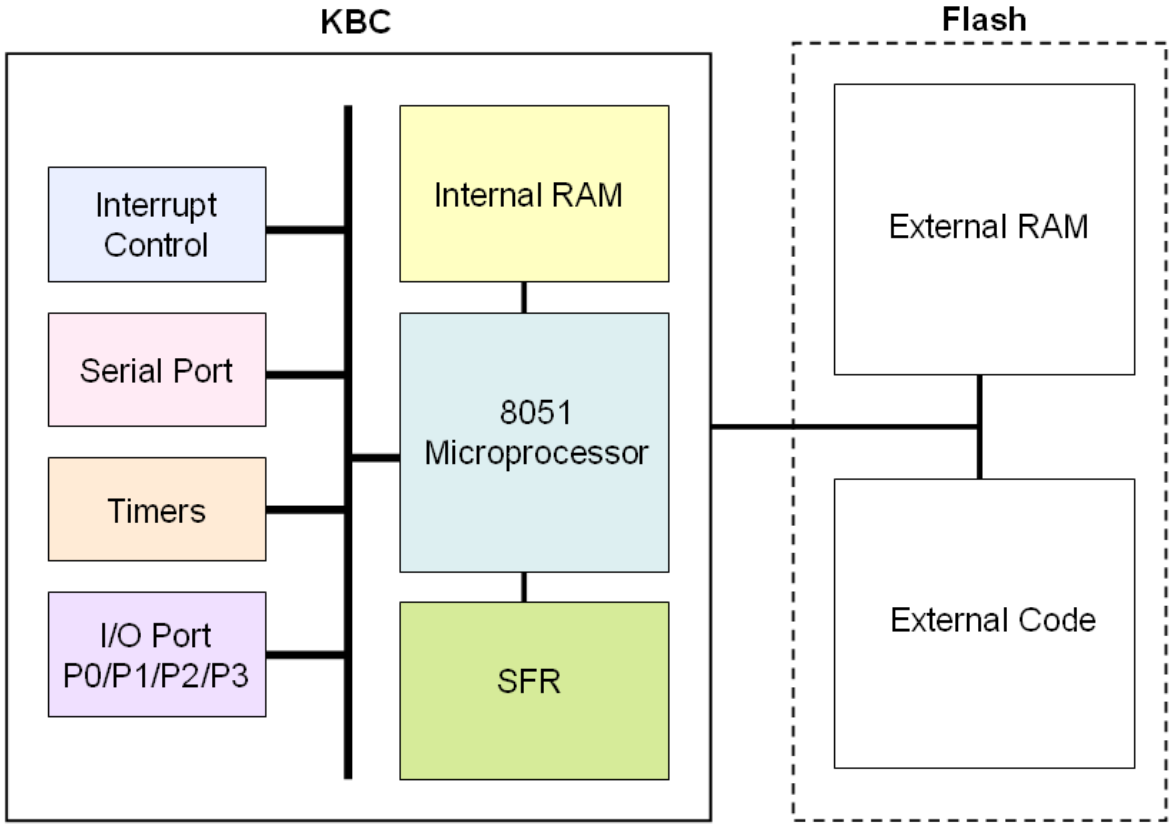
In this section gives some programming sample to control SMBus module. Please note, ENE does not guarantee these codes in every field application. The following table describes scenario of SMBus filed application.

Example
Reading status of a battery (address 0x0A)
Programming model
SMBADR (0xFF9A) = 0x0A ; battery address SMBCMD (0xFF9B) = 0x12 SMBPTCL (0xFF98) = 0x07 Wait SMBSTS (0xFF99[7]) = 1b ; command complete Check if SMBSTS[4:0] = 0000b ; no error Read SMBDAT (0xFF9C) ; the current status

4.18 8051 Microprocessor

4.18.1 8051 Microprocessor Function Description

The Microprocessor inside KBC is an industrial compatible i8051. The 8051 is featured with 128bytes Special Function Register (SFR), Serial port, 2 16-bit Timers and 3 I/O ports with interrupt capability. The 8051 operates based on external crystal and runs at 8MHz by default. The following figure gives an illustration of the 8051 architecture.



8051 Instruction

The instruction of 8051 microprocessor is fully compatible with industrial i8051. The instruction sets are as following table. The **OpCode** is in *Hexadecimal* and (b) means *Binary*. **B** stands for *byte number of instruction*. **C** stands for *number of cycle needed*.

Mnemonic	OpCode	B	C	Mnemonic	OpCode	B	C
Arithmetic				INC DPTR	A 3	1	2
ADD A, #data	2 4	2	2	MUL AB	A 4	1	2
ADD A, direct	2 5	2	2	SUBB A, #data	9 4	2	2
ADD A, @ R_0	2 6	1	2	SUBB A, direct	9 5	2	2
ADD A, @ R_1	2 7	1	2	SUBB A, @ R_0	9 6	1	2
ADD A, R_0	2 8	1	2	SUBB A, @ R_1	9 7	1	2
ADD A, R_1	2 9	1	2	SUBB A, R_0	9 8	1	2
ADD A, R_2	2 A	1	2	SUBB A, R_1	9 9	1	2
ADD A, R_3	2 B	1	2	SUBB A, R_2	9 A	1	2
ADD A, R_4	2 C	1	2	SUBB A, R_3	9 B	1	2
ADD A, R_5	2 D	1	2	SUBB A, R_4	9 C	1	2
ADD A, R_6	2 E	1	2	SUBB A, R_5	9 D	1	2
ADD A, R_7	2 F	1	2	SUBB A, R_6	9 E	1	2
ADDC A, #data	3 4	2	2	SUBB A, R_7	9 F	1	2
ADDC A, direct	3 5	2	2				
ADDC A, @ R_0	3 6	1	2	**Logic & Byte Operation**			
ADDC A, @ R_1	3 7	1	2	ANL direct, A	5 2	2	2
ADDC A, R_0	3 8	1	2	ANL direct, #data	5 3	3	2
ADDC A, R_1	3 9	1	2	ANL A, #data	5 4	2	2
ADDC A, R_2	3 A	1	2	ANL A, direct	5 5	2	2
ADDC A, R_3	3 B	1	2	ANL A, @ R_0	5 6	1	2
ADDC A, R_4	3 C	1	2	ANL A, @ R_1	5 7	1	2
ADDC A, R_5	3 D	1	2	ANL A, R_0	5 8	1	2
ADDC A, R_6	3 E	1	2	ANL A, R_1	5 9	1	2
ADDC A, R_7	3 F	1	2	ANL A, R_2	5 A	1	2
DEC A	1 4	1	2	ANL A, R_3	5 B	1	2
DEC direct	1 5	2	2	ANL A, R_4	5 C	1	2
DEC @ R_0	1 6	1	2	ANL A, R_5	5 D	1	2
DEC @ R_1	1 7	1	2	ANL A, R_6	5 E	1	2
DEC R_0	1 8	1	2	ANL A, R_7	5 F	1	2
DEC R_1	1 9	1	2	CLR A	E 4	1	2
DEC R_2	1 A	1	2	CPL A	F 4	1	2
DEC R_3	1 B	1	2	ORL direct, A	4 2	2	2
DEC R_4	1 C	1	2	ORL direct, #data	4 3	3	2
DEC R_5	1 D	1	2	ORL A, #data	4 4	2	2
DEC R_6	1 E	1	2	ORL A, direct	4 5	2	2
DEC R_7	1 F	1	2	ORL A, @ R_0	4 6	1	2
DIV AB	8 4	1	2	ORL A, @ R_1	4 7	1	2

Mnemonic	OpCode	B	C	Mnemonic	OpCode	B	C
DA A	D 4	1	2	ORL A, R_0	4 8	1	2
INC A	0 4	1	2	ORL A, R_1	4 9	1	2
INC direct	0 5	2	2	ORL A, R_2	4 A	1	2
INC @ R_0	0 6	1	2	ORL A, R_3	4 B	1	2
INC @ R_1	0 7	1	2	ORL A, R_4	4 C	1	2
INC R_0	0 8	1	2	ORL A, R_5	4 D	1	2
INC R_1	0 9	1	2	ORL A, R_6	4 E	1	2
INC R_2	0 A	1	2	ORL A, R_7	4 F	1	2
INC R_3	0 B	1	2	RL A	2 3	1	2
INC R_4	0 C	1	2	RLC A	3 3	1	2
INC R_5	0 D	1	2	RR A	0 3	1	2
INC R_6	0 E	1	2	RRC A	1 3	1	2
INC R_7	0 F	1	2	SWAP A	C 4	1	2
Logic & Byte Operation				**Data Movement**			
XRL direct, A	6 2	2	2	MOV direct, R_7	8 F	2	2
XRL direct, #data	6 3	3	2	MOV @ R_0, direct	A 6	2	2
XRL A, #data	6 4	2	2	MOV @ R_1, direct	A 7	2	2
XRL A, direct	6 5	2	2	MOV R_0, direct	A 8	2	2
XRL A, @ R_0	6 6	1	2	MOV R_1, direct	A 9	2	2
XRL A, @ R_1	6 7	1	2	MOV R_2, direct	A A	2	2
XRL A, R_0	6 8	1	2	MOV R_3, direct	A B	2	2
XRL A, R_1	6 9	1	2	MOV R_4, direct	A C	2	2
XRL A, R_2	6 A	1	2	MOV R_5, direct	A D	2	2
XRL A, R_3	6 B	1	2	MOV R_6, direct	A E	2	2
XRL A, R_4	6 C	1	2	MOV R_7, direct	A F	2	2
XRL A, R_5	6 D	1	2	MOV A, direct	E 5	2	2
XRL A, R_6	6 E	1	2	MOV A, @ R_0	E 6	1	2
XRL A, R_7	6 F	1	2	MOV A, @ R_1	E 7	1	2
				MOV A, R_0	E 8	1	2
Bit Operation				MOV A, R_1	E 9	1	2
ANL C, bit	8 2	2	2	MOV A, R_2	E A	1	2
ANL C, /bit	B 0	2	2	MOV A, R_3	E B	1	2
CLR bit	C 2	2	2	MOV A, R_4	E C	1	2
CLR C	C 3	1	2	MOV A, R_5	E D	1	2
CPL bit	B 2	2	2	MOV A, R_6	E E	1	2
CPL C	B 3	1	2	MOV A, R_7	E F	1	2
JB bit, relative	2 0	3	2	MOV direct, A	F 5	2	2
JBC bit, relative	1 0	3	2	MOV @ R_0, A	F 6	1	2
JC relative	4 0	2	2	MOV @ R_1, A	F 7	1	2
JNB bit, relative	3 0	3	2	MOV R_0, A	F 8	1	2
JNC relative	5 0	2	2	MOV R_1, A	F 9	1	2
MOV C, bit	9 2	2	2	MOV R_2, A	F A	1	2
MOV bit, C	A 2	2	2	MOV R_3, A	F B	1	2

Mnemonic	OpCode	B	C	Mnemonic	OpCode	B	C
ORL C, bit	7 2	2	2	MOV R_4, A	F C	1	2
ORL C, /bit	A 0	2	2	MOV R_5, A	F D	1	2
SETB bit	D 2	2	2	MOV R_6, A	F E	1	2
SETB C	D 3	1	2	MOV R_7, A	F F	1	2
				MOV DPTR, #data16	9 0	3	2
Data Movement				MOVC A, @ A+PC	8 3	1	>33
MOV A, #data	7 4	2	2	MOVC A, @ A+DPTR	9 3	1	>33
MOV direct, #data	7 5	3	2	MOVX A, @ DPTR	E 0	1	>=5
MOV @ R_0, #data	7 6	2	2	MOVX A, @ R_0	E 2	1	>=5
MOV @ R_1, #data	7 7	2	2	MOVX A, @ R_1	E 3	1	>=5
MOV R_0, #data	7 8	2	2	MOVX @ DPTR, A	F 0	1	>=4
MOV R_1, #data	7 9	2	2	MOVX @ R_0, A	F 2	1	>=4
MOV R_2, #data	7 A	2	2	MOVX @ R_1, A	F 3	1	>=4
MOV R_3, #data	7 B	2	2	POP direct	D 0	2	2
MOV R_4, #data	7 C	2	2	PUSH direct	C 0	2	2
MOV R_5, #data	7 D	2	2	XCH A, direct	C 5	2	2
MOV R_6, #data	7 E	2	2	XCH A, @ R_0	C 6	1	2
MOV R_7, #data	7 F	2	2	XCH A, @ R_1	C 7	1	2
MOV direct, direct	8 5	3	2	XCH A, R_0	C 8	1	2
MOV direct, @ R_0	8 6	2	2	XCH A, R_1	C 9	1	2
MOV direct, @ R_1	8 7	2	2	XCH A, R_2	C A	1	2
MOV direct, R_0	8 8	2	2	XCH A, R_3	C B	1	2
MOV direct, R_1	8 9	2	2	XCH A, R_4	C C	1	2
MOV direct, R_2	8 A	2	2	XCH A, R_5	C D	1	2
MOV direct, R_3	8 B	2	2	XCH A, R_6	C E	1	2
MOV direct, R_4	8 C	2	2	XCH A, R_7	C F	1	2
MOV direct, R_5	8 D	2	2	XCHD A, @ R_0	D 6	1	2
MOV direct, R_6	8 E	2	2	XCHD A, @ R_1	D 7	1	2
Program Branching				DJNZ direct, relative	D 5	3	2
ACALL address11	bbb1 0001	2	3	DJNZ R_0, relative	D 8	2	2
AJMP address11	bbb0 0001	2	2	DJNZ R_1, relative	D 9	2	2
CJNE A, #data, relative	B 4	3	2	DJNZ R_2, relative	D A	2	2
CJNE A, direct, relative	B 5	3	2	DJNZ R_3, relative	D B	2	2
CJNE @ R_0, #data, relative	B 6	3	2	DJNZ R_4, relative	D C	2	2
CJNE @ R_1, #data, relative	B 7	3	2	DJNZ R_5, relative	D D	2	2
CJNE R_0, #data, relative	B 8	3	2	DJNZ R_6, relative	D E	2	2
CJNE R_1, #data, relative	B 9	3	2	DJNZ R_7, relative	D F	2	2
CJNE R_2, #data, relative	B A	3	2	JMP @ A+DPTR	7 3	1	2
CJNE R_3, #data, relative	B B	3	2	JNZ relative	7 0	2	2
CJNE R_4, #data, relative	B C	3	2	JZ relative	6 0	2	2
CJNE R_5, #data, relative	B D	3	2	LCALL	1 2	3	3
CJNE R_6, #data, relative	B E	3	2	LJMP	0 2	3	2
CJNE R_7, #data, relative	B F	3	2	SJMP	8 0	2	2

Mnemonic	OpCode	B	C	Mnemonic	OpCode	B	C
RET	2 2	1	3				
RETI	3 2	1	3	**Special Instruction**			
				NOP	0 0	1	2

Interrupt Controller

In order to support more application, the 8051 in KBC extends interrupt channel to 24 for internal peripherals, that is, I/O port P0, P1 and P3 are with interrupt capability. The *interrupt priority for each channel is fixed* and no nested interrupt is supported. Here is the table to summarize the implementation of the interrupt controller.

Int. Source	Vector Address	Applications	Priority
IE0	0x0003	8051 external interrupt 0	0(Highest)
TF0	0x000B	8051 Timer 0	1
IE1	0x0013	8051 external interrupt 1	2
TF1	0x001B	8051 Timer 1	3
RI & TI	0x0023	8051 Serial port TX/RX interrupt	4
P0I[0]	0x0043	Watchdog	5
P0I[1]	0x004B	LPC I/O 0x2F R/W accessing interrupt	6
P0I[2]	0x0053	PS/2 event	7
P0I[3]	0x005B	KBC	8
P0I[4]	0x0063	IKB	9
P0I[5]	0x006B	68h/6Ch ports	10
P0I[6]	0x0073	EC	11
P0I[7]	0x007B	ESB events	12
P1I[0]	0x0083	FAN0 monitor event (update/overflow)	13
P1I[1]	0x008B	FAN1 monitor event (update/overflow)	14
P1I[2]	0x0093	SMBus events	15
P1I[3]	0x009B	CIR events	16
P1I[4]	0x00A3	GPT0 event	17
P1I[5]	0x00AB	GPT1 event	18
P1I[6]	0x00B3	GPT2 event	19
P1I[7]	0x00BB	GPT3 event	20
P3I[0]	0x00C3	Write extended I/O (LPC I/O port 80)	21
P3I[1]	0x00CB	GPIO00~GPIO0F	22
P3I[2]	0x00D3	GPIO10~GPIO1F	23
P3I[3]	0x00DB	GPIO20~GPIO2F	24
P3I[4]	0x00E3	GPIO30~GPIO3F	25
P3I[5]	0x00EB	GPIO40~GPIO4F	26
P3I[6]	0x00F3	GPIO50~GPIO59 / GPXIOD00~GPXIOD07	27
P3I[7]	0x00FB	ADC update	28(Lowest)

Interrupt Enable/Flag Table

	Interrupt Enable			Pending Flag		
	address	bit	behavior	address	bit	type
8051 external interrupt0 (GPIO1A)	A8h	0	2	88h	1	2
8051 Timer0	A8h	1	2	88h	5	2
8051 external interrupt0 (GPIO1B)	A8h	2	2	88h	3	2
8051 Timer1	A8h	3	2	88h	7	2
8051 Serial Port	A8h	4	2	98h	1~0	1
WDT	FE80h	1	1	FE81h	1	1
				FE81h	0	1
RTC	FE84h	7,0	1	FE84h	1	1
LPC I/O R/W 0x2F	FF20h	2	1	-	-	-
	FF9Ah	1	1	FE9Ah	2	1
PS/2	FEE0h	3~0	2	FEE1h	3~0	2
KBC	FC81h	1,0	1	FC82h	1,0	2
IKB	FCA3h	5~0	1	FCA4h	5~0	2
LPC 68h/6Ch IBF_Rising OBF_Falling	FE9Dh	1,0	1	FE9Eh	1,0	1
				FE9Eh	3,2	2
EC host interrupt	FF04h	1	4	FF1Ah	1	1
	FF04h	0	2	FF1Ah	0	2
ESB	FC90h	2	3	-	-	-
	FC92h	6~4	3	FC91h	6~4	1
	FC92h	3~0	3	FC97h	7~0	1
	FC98h	7~4	3	FC97h		1
FAN	FE20h	3,2	3	FE21h	1,0	1
	FE30h	3,2	3	FE31h	1,0	1
SMBus	FF95h	0	1	FF99h	7,5	1
				FF96h	5	1
	FF95h	1	1	FF99h	6	1
	FF95h	2	1	FF96h	3	1
CIR TX	FEC0h	5	1	FEC2h	3	1
CIR RX	FEC0h	1	1	FEC2h	2~0	1
GPT0~GPT3	FE50h	3~0	1	FE51h	3~0	2
Wrute Extended I/O	FE95h	4	1	-	-	-
GPWU	FF3Xh	7~0	3	FF4Xh	7~0	1
ADC	FF18h	0	1	-	-	-

behavior	Interrupt Behavior => (Interrupt Occurs) (1) IE bit = 1, interrupt asserts when trigger event occurs (2) IE bit = 1, interrupt asserts when trigger event occurs but if PF not clear, interrupt will continue asserting (3) IE = 1, interrupt asserts when trigger event occurs

	or IE bit is from low to high(0 -> 1) when Pending Flag(PF) is = 1 (4) No matter IE bit = 1 or 0, interrupt asserts when trigger event occurs
type	Pending Flag(PF) => 6. When trigger event occurs, PF will be set to 1. PF cleared to 0 by WC1/WC0 (2) IE bit = 1, when event occurs, PF will be set to 1. PF is cleared to 0 by WC1/WC0
Interrupt Enable = (IE) Pending Flag = (PF) WC1 = Write 1 clear WC0 = Write 1 clear	

8051 Special Function Register (SFR)

The Special Function Registers are located in the internal RAM of 8051 microprocessor. The internal address are from 0x80 to 0xFF, sized with 128 bytes. All the SFRs are compatible with the standard ones. Just few SFRs are redesigned with new features for flexible application. The following table gives a brief summary. For more detail, please refer to the section of register description.

80	P0IE	SP	DPL	DPH			PCON2	PCON	87
88	TCON	TMOD	TL0	TL1	TH0	TH1			8F
90	P1IE								97
98	SCON	SBUF	SCON2	SCON3					9F
A0	P2								A7
A8	IE								AF
B0	P3IE								B7
B8	IP								BF
C0									C7
C8									CF
D0	PSW								D7
D8	P0IF								DF
E0	ACC								E7
E8	P1IF								EF
F0	B								F7
F8	P3IF								FF
	★								

1. The blue parts are changed from standard features and the green ones are the new design for special features. And all the others are the standard features of conventional 8051.
2. The registers listed in the column with ★mark are all *bit addressable*.

8051 Microprocessor Register Description

The SFR registers are located at internal RAM 0x80 ~ 0xFF.

P0 Interrupt Enable Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x80	P0IE	7-0	R/W	P0 interrupt enable. Bit0~bit7 stand for corresponding P0[0]~P0[7]. **0**: Disable **1**: Enable	0x00

Stack Pointer					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x81	SP	7-0	R/W	8051 stack pointer register	0x07

Data Pointer Low Byte					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x82	DPL	7-0	R/W	Low byte of DPTR	0x00

Data Pointer High Byte					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x83	DPH	7-0	R/W	High byte of DPTR	0x00

Reserved					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x84	RSV	7-0	RSV	Reserved	0x00

Reserved					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x85	RSV	7-0	RSV	Reserved	0x00

Processor Control Register 2					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x86	PCON2	7	R/W	Reserved but this bit should be "0".	0x20
		6	R/W	Timer0/Timer1 test mode enable. **0**: Disable **1**: Enable	
		5	R/W	Reserved	
		4	R/W	KBC modules write control. Once this bit set, 8051 could issue write access to external modules. **0**: Disable **1**: Enable	
		3	R/WC0	Same interrupt source pending flag. If the 8051 is handling some interrupt event, at the same time, the same source asserting the interrupt again, this flag will be set. If this flag set, the 8051 will re-enter ISR again once executing IRET. Writing "**0**" to clear this flag.	
		2-1	RSV	Reserved	
		0	R/W	Not fetching instruction while in idle loop. **0**: Disable **1**: Enable	

Processor Control Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x87	PCON	7-6	RSV	Reserved	0x00
		5	R/W	Interrupt vector offset address1 **0**: Interrupt vector address offset adding 0x0 **1**: Interrupt vector address offset adding 0x8000	
		4	R/W	Interrupt vector offset address2 **0**: Interrupt vector address offset adding 0x0 **1**: Interrupt vector address offset adding 0x4000 Please note, if PCON[5]=1 and PCON[4]=1 then the result of interrupt vector address will be added 0xC000.	
		3	R/W	General purpose flag 1 **0**: no event **1**: event occurs	
		2	R/W	General purpose flag 2 **0**: no event **1**: event occurs	
		1	WO	Stop mode enable. All clock stop except the external 32.768K OSC and PCICLK. **1**: Enable (write "0" no work)	
		0	WO	Idle mode enable. The clock of 8051 stops. **1**: Enable (write "0" no work)	

Timer/Counter Control Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x88	TCON	7	R/W1C	**TF1**, Timer1 overflow flag **0**: no event **1**: event occurs	0x00
		6	R/W	**TR1**, Timer1 start control. **0**: stop to count **1**: start to count	
		5	R/W1C	**TF0**, Timer0 overflow flag **0**: no event **1**: event occurs	
		4	R/W	**TR0**, Timer0 start control. **0**: stop to count **1**: start to count	
		3	R/W1C	**IE1**, External interrupt 1 flag **0**: no event **1**: event occurs	
		2	R/W	**IT1**, External interrupt 1 trigger selection **0**: low level trigger **1**: falling edge trigger	
		1	R/W1C	**IE0**, External interrupt 0 flag **0**: no event **1**: event occurs	
		0	R/W	**IT0**, External interrupt 0 trigger selection **0**: low level trigger **1**: falling edge trigger	

Timer Mode Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x89	TMOD	7	R/W	**GATE1**, this bit is the gate control of TR1 and INT1 **0**: Disable **1**: Enable	0x00
		6	R/W	**CT1**, Timer1 timer/counter selection **0**: Timer **1**: Counter	
		5-4	R/W	**TM1**, Timer1 mode selection **0**: 13-bit timer **1**: 16-bit timer **2**: 8-bit auto reload timer **3**: Timer 1 stops.	
		3	R/W	**GATE0**, this bit is the gate control of TR0 and INT0 **0**: Disable **1**: Enable	
		2	R/W	**CT0**, Timer0 timer/counter selection **0**: Timer **1**: Counter	
		1-0	R/W	**TM0**, Timer0 mode selection **0**: 13-bit timer **1**: 16-bit timer **2**: 8-bit auto reload timer **3**: TL0 and TH0 are two 8-bit timers.	

Timer 0 Low Byte					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x8A	TL0	7-0	R/W	Low byte of timer 0	0x00

Timer 1 Low Byte					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x8B	TL1	7-0	R/W	Low byte of timer 1.	0x00

Timer 0 High Byte					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x8C	TH0	7-0	R/W	High byte of timer 0	0x00

Timer 1 High Byte					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x8D	TH1	7-0	R/W	High byte of timer 1	0x00

Port1 Interrupt Enable Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x90	P1IE	7-0	R/W	Port 1 interrupt enable. Bit0~Bit7 stand for P1[0]~P1[7] respectively **0**: Disable **1**: Enable	0x00

Reserved					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x91-0x97	N/A	7-0	RSV	Reserved	0x00

Serial Port Control Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x98	SCON	7-6	R/W	**SM1,SM0**, serial port mode **00**: 8-bit shift register, E51RX will be shift clock of E51CLK. **01**: 8-bit serial port (variable) **10**: 9-bit serial port (variable) **11**: 9-bit serial port (variable)	0x50
		5	RSV	Reserved	
		4	R/W	**REN**, serial port receive function enable. **0**: Disable **1**: Enable	
		3	R/W	**TB8**, The 9th bit of transmit data in mode2 and mode3.	
		2	R/W	**RB8**, The 9th bit of receive data	
		1	R/W0C	**TI**, TX interrupt flag **0**: no event **1**: event occurs	
		0	R/W0C	**RI**, RX interrupt flag **0**: no event **1**: event occurs	

Serial Port Data Buffer Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x99	SBUF	7-0	R/W	Serial port data buffer	0x00

Serial Port Control Register 2					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x9A	SCON2	7-0	R/W	High byte of 16-bit counter for baud rate	0x00

Serial Port Control Register 3					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0x9B	SCON3	7-0	R/W	Low byte of 16-bit counter for baud rate	0x00

Port 2 Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0xA0	P2	7-0	R/W	Port 2 register	0x00

Interrupt Enable Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0xA8	IE	7	R/W	**EA**, all interrupts enable. **0**: Disable **1**: Enable	0x00
		6-5	RSV	Reserved	
		4	R/W	**ES**, serial port interrupt enable **0**: Disable **1**: Enable	
		3	R/W	**ET1**, timer1 overflow interrupt enable **0**: Disable **1**: Enable	
		2	R/W	**EX1**, external interrupt 1 enable. **0**: Disable **1**: Enable	
		1	R/W	**ET0**, timer0 overflow interrupt enable **0**: Disable **1**: Enable	
		0	R/W	**EX0**, external interrupt 0 enable. **0**: Disable **1**: Enable	

Interrupt Enable Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0xB0	P3IE	7-0	R/W	Port 3 interrupt enable. Bit0~Bit7 stand for P3[0]~P3[7] respectively **0**: Disable **1**: Enable	0x00

Interrupt Priority Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0xB8	IP	7-5	RSV	Reserved	0x00
		4	R/W	Serial port interrupt priority **0**: Low **1**: High	
		3	R/W	Timer1 interrupt priority **0**: Low **1**: High	
		2	R/W	External interrupt 1 priority **0**: Low **1**: High	
		1	R/W	Timer 0 interrupt priority **0**: Low **1**: High	
		0	R/W	External interrupt 0 priority **0**: Low **1**: High	

Processor Status Word Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0xD0	PSW	7	R/W	**CY**, carry flag	0x00
		6	R/W	**AC**, auxiliary carry flag.	
		5	R/W	**F0**, for user general purpose.	
		4	R/W	**RS1**, register bank selector 1.	
		3	R/W	**RS0**, register bank selector 0.	
		2	R/W	**OV**, overflow flag	
		1	R/W	**F1**, flag 1 for user general purpose	
		0	R/W	**P**, parity flag	

Port0 Interrupt Flag Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0xD8	P0IF	7-0	R/W	Port 0 interrupt flag.	0x00

Accumulator, ACC					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0xE0	ACC	7-0	R/W	Accumulator	0x00

Port1 Interrupt Flag Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0xE8	P1IF	7-0	R/W	Port 1 interrupt flag.	0x00

B Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0xF0	B	7-0	R/W	**B** register, for MUL and DIV instructions.	0x00

Port3 Interrupt Flag Register					
Address	**Name**	**Bit**	**Type**	**Description**	**Default**
0xF8	P3IF	7-0	R/W	Port 3 interrupt flag.	0x00

6. Electrical Characteristics

5.1 Absolute Maximum Rating

Symbol	Parameter	Condition	Rating	Unit
V_{CC}	Power Source Voltage	All voltages are referred to GND.	-0.3 ~ 3.6	V
V_i	Input Voltage		-0.3 ~ 3.6	V
V_o	Output Voltage		-0.3 ~ 3.6	V
T_{STG}	Storage Temperature		-65 ~ 150	℃
	ESD Tolerance	C_{ZAP} = 100pF R_{ZAP} = 1.5KΩ		V

5.2 DC Electrical Characteristics

Parameter	Symbol	Min	Typ.	Max	Unit	Condition
Input Low Voltage	V_{IL}	-0.3		V_{CC}*0.3	V	V_{CC}=3.0~3.6V
Input High Voltage	V_{IH}	V_{CC}*0.7		V_{CC}*1.1	V	V_{CC}=3.0~3.6V
Input High Voltage(5V Tolerant)	V_{IH}	V_{CC}*0.7		5.5	V	V_{CC}=3.0~3.6V
Output Low Voltage	V_{OL}		0.4		V	
Output High Voltage	V_{OH}		2.8		V	
Input Low Threshold	V_{t-}		1.14		V	
Input High Threshold	V_{t+}		1.94		V	
Hysteresis	V_{TH}		0.8		V	
Input Leakage Current	I_{IL}		1		μA	No pull-up
Tri-state Leakage Current	I_{OZ}		1		μA	No pull-up
Input Pull-Up Resistance	R_{PU}	30K	40K	50K	Ω	V_I=0V
Input Capacitance	C_{PU}		5.5		pF	
Output Capacitance	C_{OUT}		5.5		pF	
Bi-directional Capacitance	C_{BID}		5.5		pF	

5.3 A/D Characteristics

Parameter	Limits			Unit
	Min	Typ	Max	
Resolution		10		Bit
Integral Non-linearity Error (INL)		±2		LSB
Differential Non-linearity Error (DNL)		±1		LSB
Offset Error		±1		LSB
Gain Error		±1		LSB
A/D Input Voltage Range	$0.1V_{cca}$		$0.9V_{cca}$	V
A/D Input Leakage Current		<0.1		uA
A/D Input Resistance	10			MΩ
A/D Input Capacitance			8	pF
A/D Clock Frequency		1		MHz
Voltage Conversion Time		3		us

5.4 Recommend Operation Condition

Symbol	Parameter	Limits			Unit
		Min	Typ	Max	
Vcc	Power Source Voltage	3.0	3.3	3.6	V
GND	Ground Voltage	-0.3	0	0.3	V
V_{CCA}	Analog Reference Voltage (for A/D and D/A)	3.0	3.3	3.6	V
AGND	Analog Ground Voltage		0		V
T_{op}	Operating Temperature	0	25	70	℃

5.5 Operating Current

Symbol	Parameter	Limits	Unit
		Typ	
Icc	Typical current consumption in operating state under Windows environment. All clock domains are running, and no keyboard/mouse activities.	20	mA

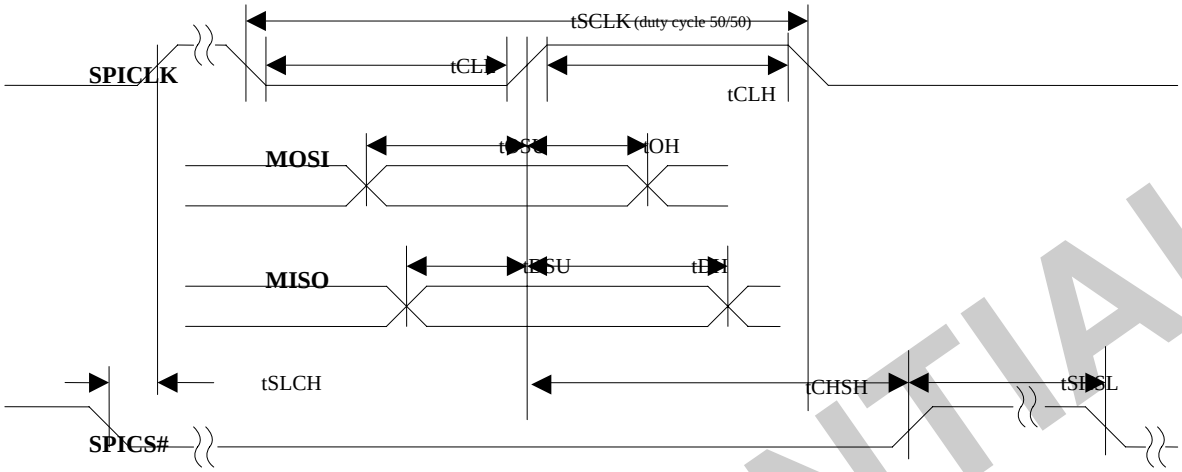
5.6 Package Thermal Information

Thermal resistance (degrees C/W). $Theta_{JA}$ values for KB926D.

	$Theta_{JA}$ @ 0 m/s	**$Theta_{JA}$ @ 1 m/s**	**$Theta_{JA}$ @ 2 m/s**
128-Pin LQFP	59.1	53.3	51.4
128 LFBGA	50.9	48.7	45.5

5.7 AC Electrical Characteristics

5.7.1 SPI Flash Timing



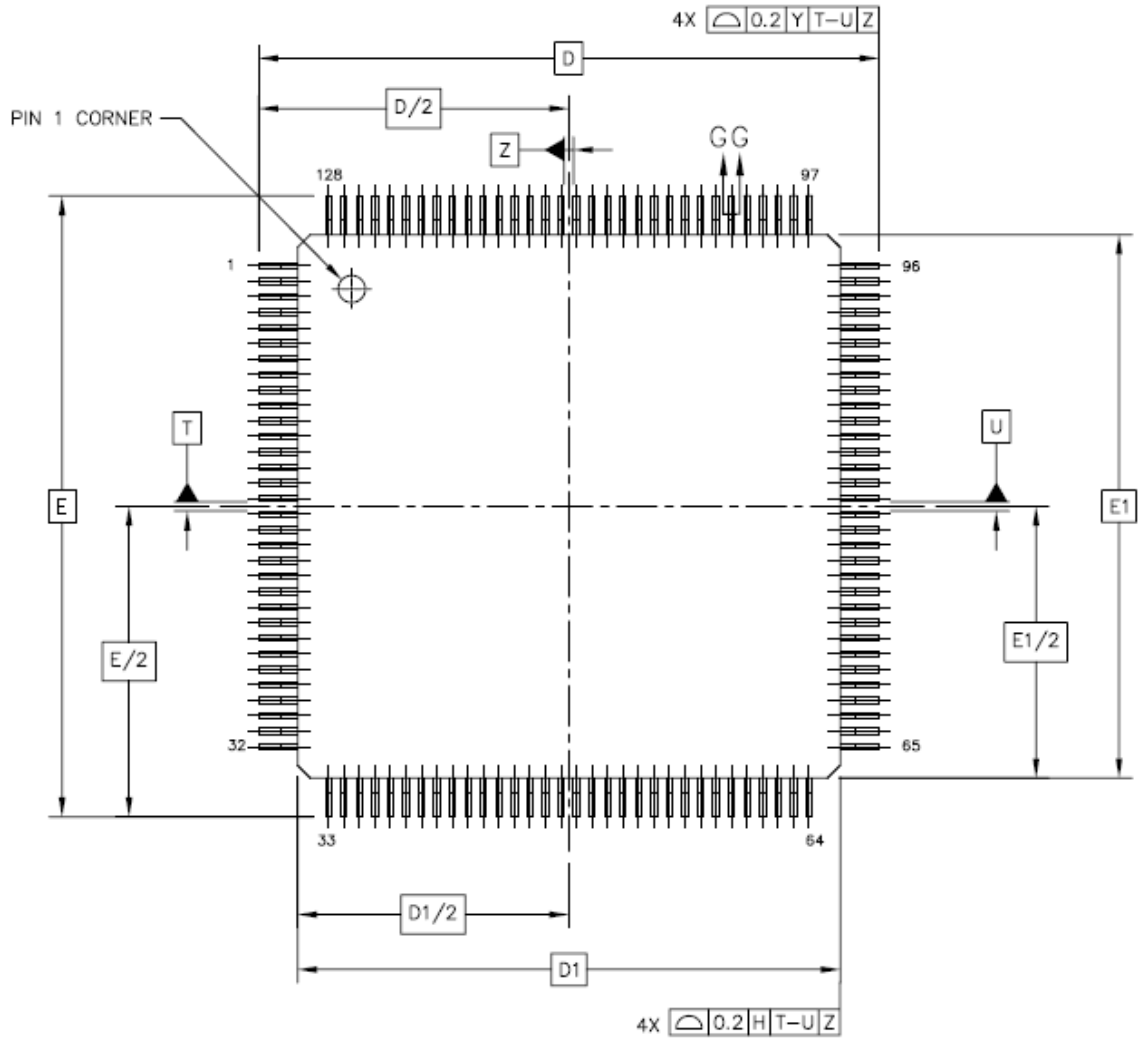
Parameter	Symbol	Spec.				Condition
		Min.	Typ.	Max.	Unit	
SPICLK period	**tSCLK**	15.2[*1]	--	--	ns	
SPICLK High Period	**tCLH**	4	--	--	ns	**tSCLK** = 15.2ns,
SPICLK Low Period	**tCLL**	4	--	--	ns	**tSCLK** = 15.2ns,
MOSI Setup Time	**tOSU**	--	tSCLK/2 – 5[*2]	--	ns	
MOSI Hold Time	**tOH**	--	tSCLK/2 + 5[*2]	--	ns	
SPICS# Active Setup Time	**tSLCH**	tSCLK	1.5 * tSCLK[*3]	--	ns	
SPICS# Not Active Hold Time	**tCHSH**	0.5 * tSCLK -5	0.5 * tSCLK[*3]	--	ns	
SPICS# Deselect Time	**tSHSL**	2 * tSCLK	--	--	ns	
MISO Setup Time	**tDSU**	0	--	--	ns	
MISO Hold Time	**tDH**	tSCLK/2 – 4	--	--	ns	

1. Tolerance +/- 3% (need to count in the DPLL tolerance)
2. For characteristic only
3. Not fully tested, depending on different application circuit,
4. All pin loading is based on 12pf simulation

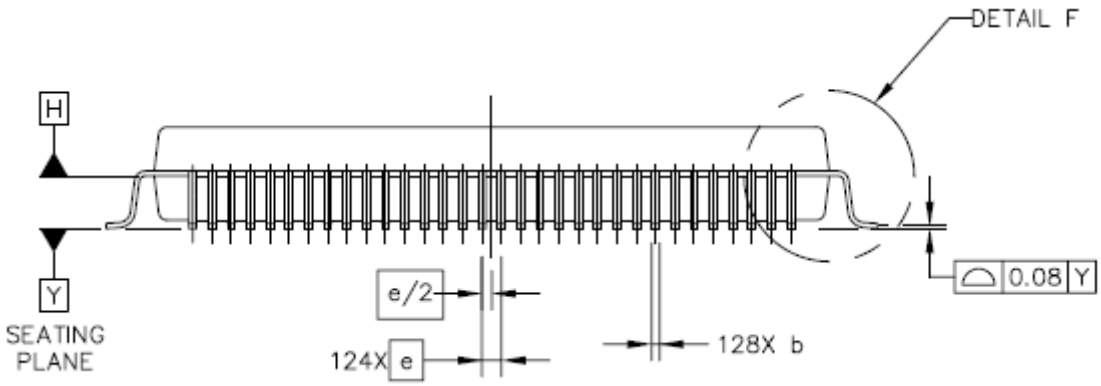
6. Package Information

6.1 LQFP 128-Pin Outline Diagram

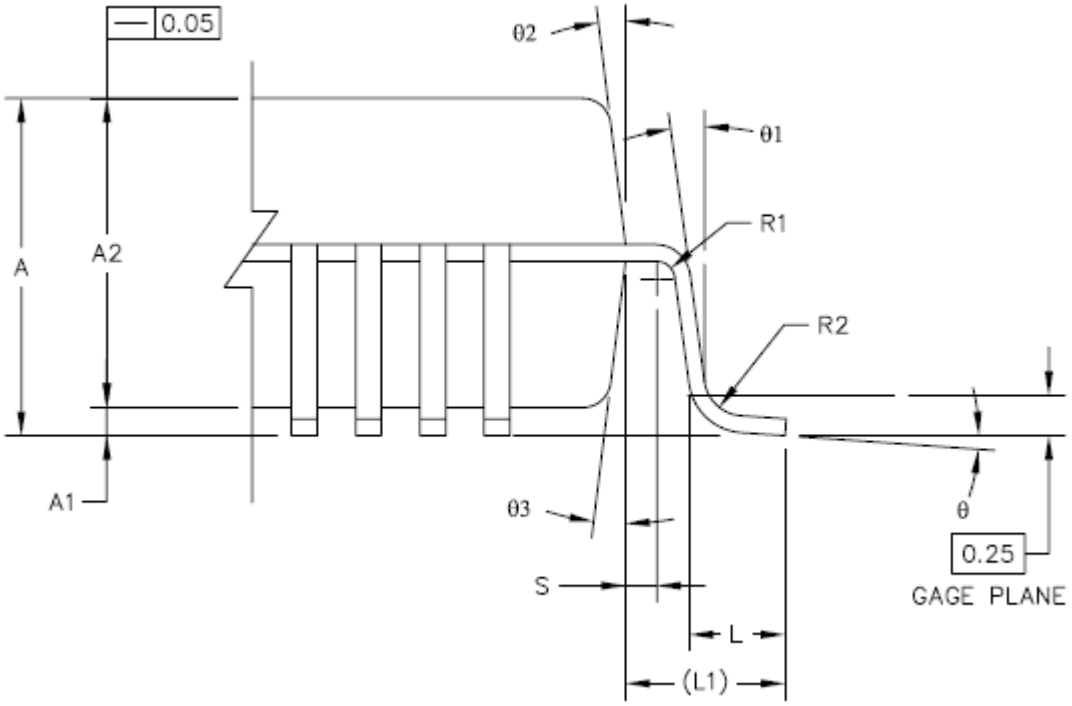
6.1.1 Top View

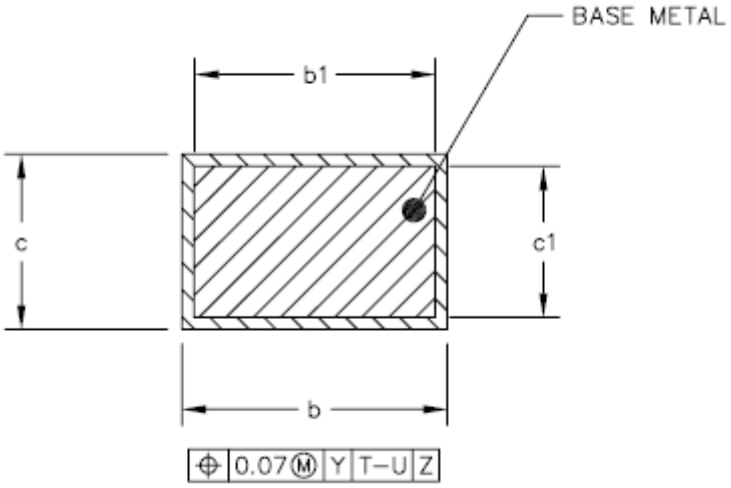


6.1.2 Side View



6.1.3 Lead View



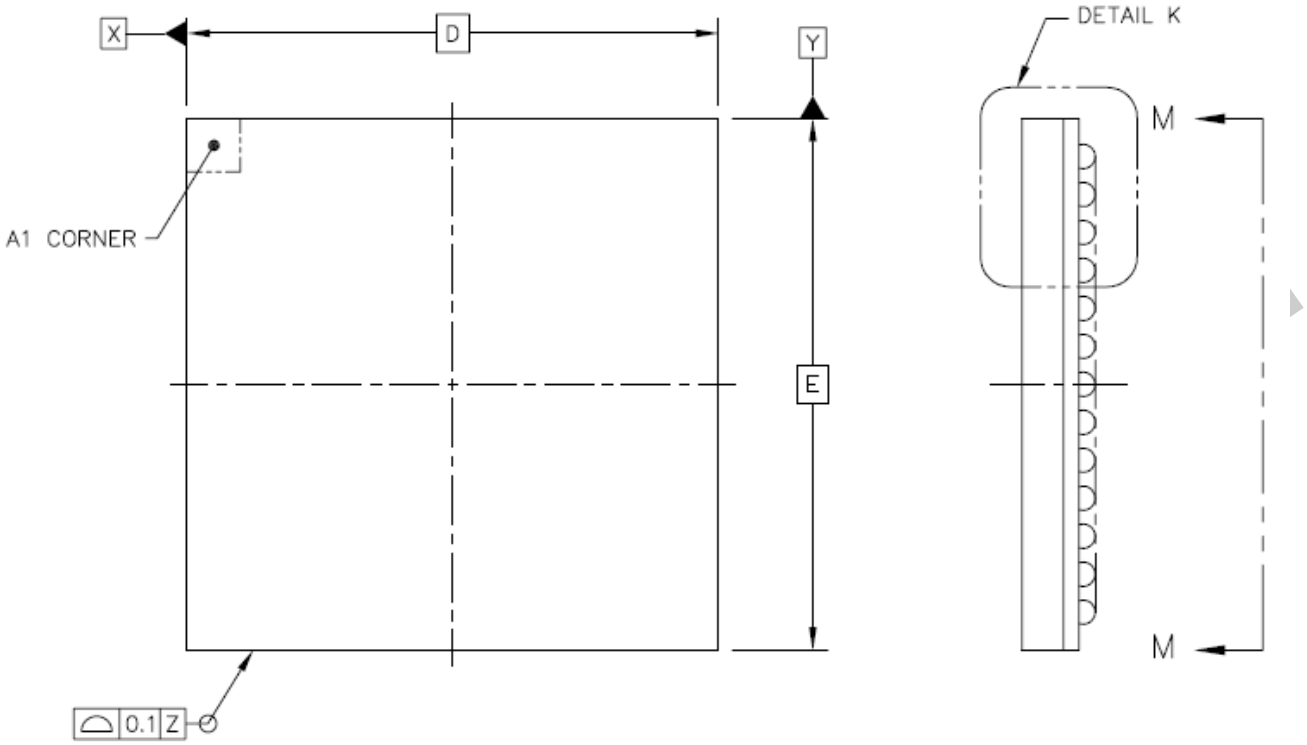


6.1.4 LQFP Outline Dimensions

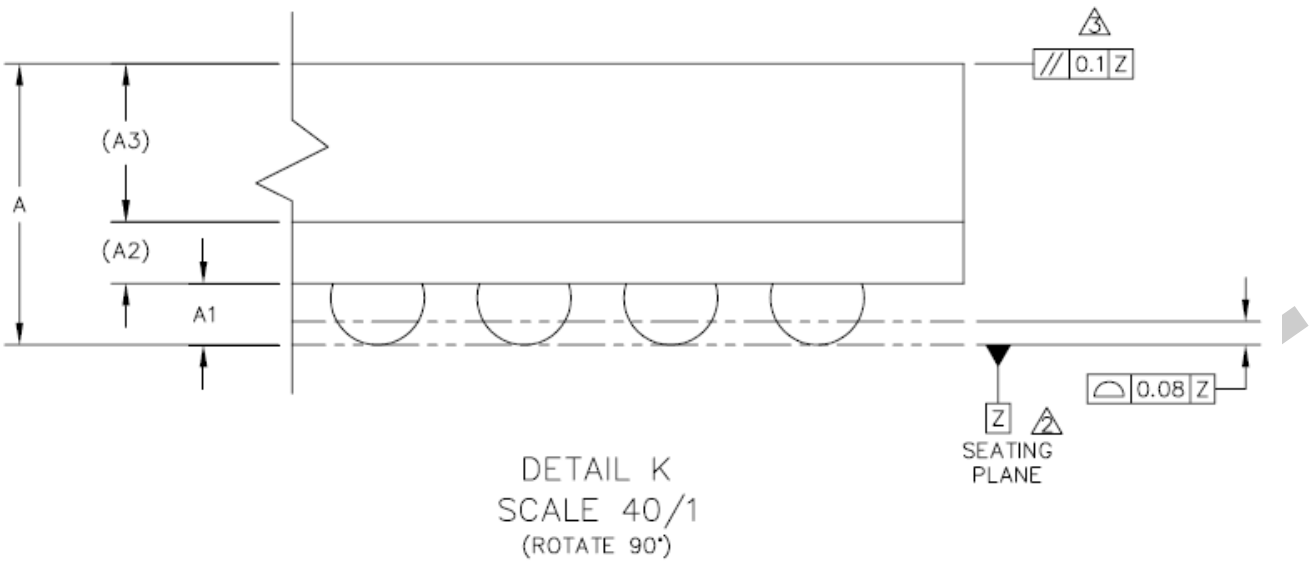
DIM	Min.	Typ.	Max.	DIM	Min.	Typ.	Max.
A	———		1.6	E1		14 BSC	
A1	0.05		0.15	L	0.45	0.6	0.75
A2	1.35	1.4	1.45	L1		1 REF	
b	0.13	0.16	0.23	R1	0.08		———
b1	0.13		0.19	R2	0.08		0.2
c	0.09		0.2	S	0.2		———
c1	0.09		0.16	θ	0°	3.5°	7°
D		16 BSC		θ1	0°		———
D1		14 BSC		θ2	11°	12°	13°
e		0.4 BSC		θ3	11°	12°	13°
E		16 BSC					
Unit	mm						
Package	14x14x1.4						
Pitch POD	0.4						
Footprint	2mm						

6.2 LFBGA 128-Pin Outline Diagram

6.2.1 Top View

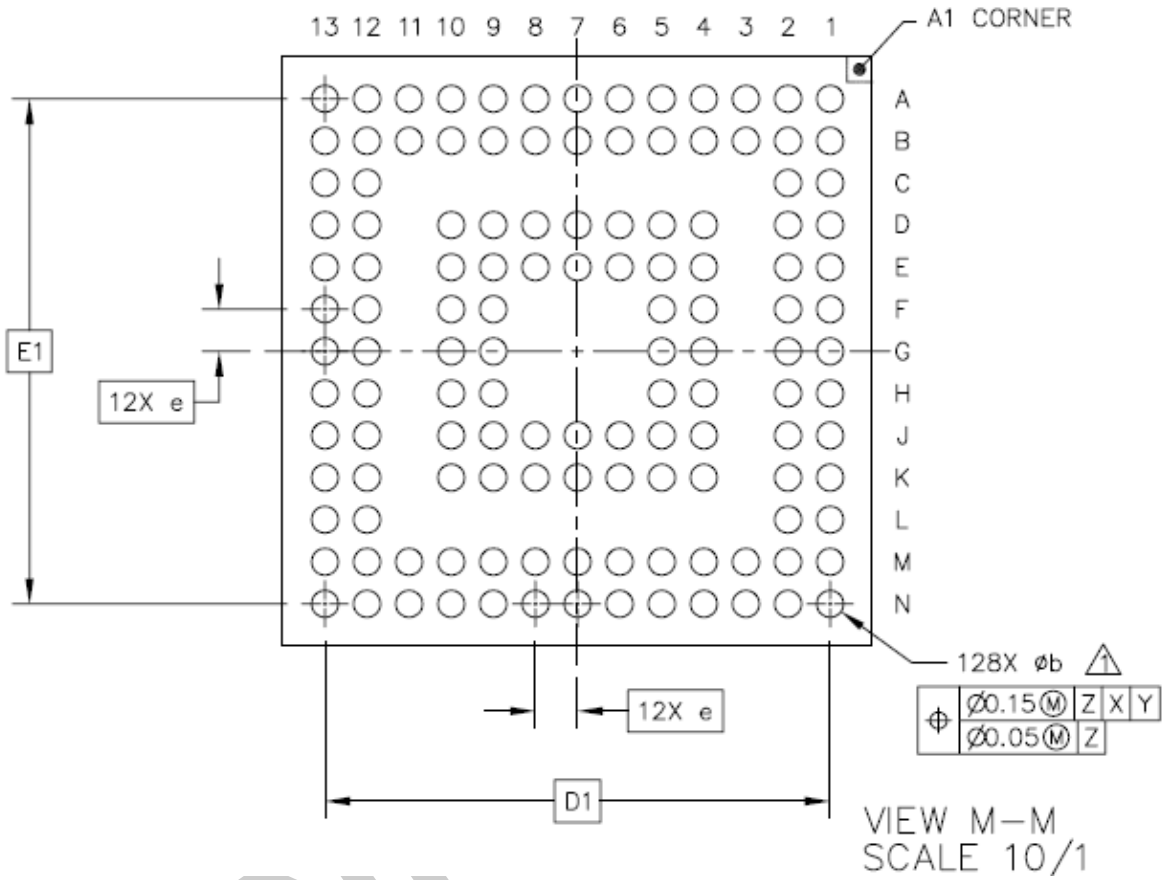


6.2.2 Side View



DETAIL K
SCALE 40/1
(ROTATE 90°)

6.2.3 Bottom View



6.2.4 LFBGA Outline Dimensions

DIM	Min.	Nor.	Max.
A	———		1.3
A1	0.16		0.26
A2		0.21	
A3		0.7	
b	0.27		0.37
D		7	
E		7	
e		0.5	
D1		6	
E1		6	
Unit	mm		
Package	7mm * 7 mm		

6.3 Part Number Description

Part Number	Package Size	Lead Free Process
KB926QF D2	14mm * 14mm 128 pins LQFP	Lead Free
KB926QF D3	14mm * 14mm 128 pins LQFP	Lead Free
KB926BF D3	7mm * 7mm 128 balls LFBGA	Lead Free